



A University of Sussex DPhil thesis

Available online via Sussex Research Online:

<http://sro.sussex.ac.uk/>

This thesis is protected by copyright which belongs to the author.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Please visit Sussex Research Online for more information and further details



University of Sussex

An Ant-Inspired, Deniable Routing Approach in Ad Hoc Question & Answer Networks.

Simon Alexis Fleming

The Foundations of Software Systems Group
School of Informatics
University of Sussex

A thesis submitted, on the 30th of September 2011, in partial fulfilment of the requirements for the degree of Doctor of Philosophy (DPhil) in the School of Informatics at the University of Sussex.

To my family and close friends.

*What's the use of two strong legs
if you only run away
and what's the use of the finest voice
if you've nothing good to say
what's the use in strength and muscle
if you only push and shove
and what's the use of two good ears
if you can't hear those you love*

Eddie Reader

What you do with what you've got (1993)

Statement of Originality

This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and the Acknowledgments.

This thesis is not substantially the same as any that I have submitted or am currently submitting for a degree, diploma or any other qualification at any other university. No part of this dissertation has already been, or is being, submitted for any such degree, diploma or qualification.

30th of September 2011

© 2011 University of Sussex.

All trademarks used in this thesis are hereby acknowledged.

Abstract

The ubiquity of the Internet facilitates electronic question and answering (Q&A) between real people with ease via community portals and social networking websites. It is a useful service which allows users to appeal to a broad range of answerers. In most cases however, Q&A services produce answers by presenting questions to the general public or associated digital community with little regard for the amount of time users spend examining and answering them. Ultimately, a question may receive large amounts of attention but still not be answered adequately.

Several existing pieces of research investigate the reasons why questions do not receive answers on Q&A services and suggest that it may be associated with users being afraid of expressing themselves. Q&A works well for solving information needs, however, it rarely takes into account the privacy requirements of the users who form the service.

This thesis was motivated by the need for a more targeted approach towards Q&A by distributing the service across ad hoc networks. The main contribution of this thesis is a novel routing technique and networking environment (distributed Q&A) which balances answer quality and user attention while protecting privacy through plausible deniability. Routing approaches are evaluated experimentally by statistics gained from peer-to-peer network simulations, composed of Q&A users modelled via features extracted from the analysis of a large Yahoo! Answers dataset. Suggestions for future directions to this work are presented from the knowledge gained from our results and conclusions.

Acknowledgments

This work would have never been completed without the support of my parents, siblings and friends.

Particular thanks to my supervisors Dr. Dan Chalmers and Dr. Ian Wake-man for their invaluable guidance, patience and support over the past three years.

I would also like to extend my thanks to other members and friends of the Foundations of Software Systems Group, namely: Dr. Des Watson, Dr. Anirban Basu, Raphael Commins, Dr. Jon Robinson, Ben Horsfall, Tom Harvey, Danny Mathews, Dr. Martin Berger, Dr. Eskindir Asmare, Dr. Billiejoe Charlton, Dr. Bernhard Reus, Dr. John A. Carroll and Renan Krishna for their help, guidance, support and intellectual conversations throughout the duration of my DPhil at the University of Sussex.

I would like to thank my friends: Ben Mottram, Chris Aplin, Jamie Philpin and Owain Lloyd for all the support and encouragement they have provided me along the way.

A very big Thank you to my family: Mum, Dad, Kelly and Oliver. Especially my mother who provided me with shelter and support in her lovely home in leafy Surrey.

I would like to remember my late father, who having battled cancer for over a year, didn't quite make it to my graduation. You were very brave.

Thank you all!

Table of Contents

Statement of Originality	2
Abstract	3
Acknowledgments	4
List of Figures	12
List of Tables	17
1 Introduction	18
1.1 Contribution	20
1.2 Thesis Structure	20
1.3 Published Work	22
2 Literature Review	23
2.1 The Internet and the World Wide Web	23
2.1.1 Mobile ad hoc Networks	24
2.1.1.1 Internet-Based Mobile ad hoc Networks	25
2.1.2 The Growth of User Generated Content	26
2.1.2.1 Twitter	27
2.1.3 Knowledge Markets and Public Search	29
2.1.4 Information Overload	29
2.2 Question and Answer Services	31
2.2.1 Yahoo! Answers	33
2.2.2 Aardvark	36
2.2.3 Real-Time Question and Answer Services	39
2.2.4 Expert Retrieval	42
2.3 Digital Security and Privacy of Communication	44

2.3.1	The Internet	44
2.3.2	Anonymous Communication Networks	46
2.3.3	Sender Anonymity	47
2.3.4	Receiver Anonymity	48
2.3.5	Location Privacy	48
2.3.6	Anonymous Opinion Exchange	49
2.3.7	Crowds: Anonymity for Web Transactions	50
2.4	Swarm Intelligence	51
2.4.1	Stigmergy	51
2.4.2	Foraging Strategies in Ants	52
2.4.3	Key Elements of the Stigmergic Technique	55
2.4.4	Security and Privacy	55
2.4.5	Telecommunication Stigmergy	56
2.4.6	Stigmergy in Computer Networks	58
2.4.7	Stigmergic Technique Conclusions	62
2.5	Peer-to-Peer	62
2.5.1	Peer-to-Peer Networking	63
2.5.1.1	Neighbours and Connectivity	63
2.5.1.2	Network Churn	64
2.5.1.3	Bootstrapping	64
2.5.1.4	Routing Tables	65
2.5.2	Example Peer-to-Peer Applications	65
2.5.3	Advantages and Disadvantages	67
2.5.4	Routing in P2P Networks	67
2.5.5	Peer-to-Peer Simulation	69
2.6	Summary and Research Direction	70
3	User Modelling	71
3.1	Background	71
3.2	Yahoo! Answers dataset	71
3.2.1	Database Structure	72
3.2.2	Languages	74
3.2.3	Date Ranges	74
3.2.4	Corpus Issues and Discrepancies	75
3.2.5	Category Popularity	76
3.2.6	Interests and Expertise	77

3.2.7	Expertise Levels	79
3.2.8	Question and Answer Lengths	81
3.3	Twitter Corpus	82
3.3.1	Twitter Application Programming Interface	83
3.3.2	Bespoke Twitter Crawler	83
3.3.3	Timezones	86
3.3.4	Number of Active Days	87
3.3.5	Update Frequencies	88
3.3.6	Inter-Tweet (Repeat Activity Times)	88
3.3.7	Inter-tweet Reply Time (Response Times)	90
3.3.8	Inter-Tweet Question Time (Question Intervals)	90
3.3.9	Tweet Replies (Popularity)	91
3.3.10	Conversation Lengths (Chains of Interaction)	92
3.4	User Task Processing	93
3.4.1	User Attention	94
3.4.2	User States	95
3.4.3	Composing and Reading	96
3.5	User Churn Models	97
3.5.1	The Weibull Distribution	98
3.5.2	Possible Churn Scenarios	100
3.6	Probabilistic Modelling	100
3.7	Summary	101
4	Deniable Routing for Q&A	102
4.1	Homogeneous Network Topology	102
4.2	Question Routing	103
4.2.1	Naïve Routing Approaches	103
4.2.1.1	Flooding	103
4.2.1.2	Random Hops	104
4.3	Stigmergic Routing	105
4.3.1	Overview of Technique	106
4.3.2	Question Routing Protocol	106
4.3.3	Answer Protocol	107
4.3.4	Feedback Protocol	108
4.3.5	Routing Tables and Pheromones	108
4.3.6	Pheromone Update Rules	109

4.3.7	Probabilistic Path Selection	111
4.3.8	Routing Variations	114
4.3.9	Learning and Warm-up Periods	117
4.3.10	Network Churn	118
4.3.11	Pheromone Evaporation	119
4.3.12	Pheromone Defaults, Maxima and Minima	119
4.3.13	Oracle Nodes	119
4.4	Attack Models	120
4.4.1	Establishing Author Identity	120
4.4.2	Reducing Answer Quality	121
4.4.3	Eager Answerer	121
4.4.4	Denial of Service (DoS)	122
4.4.5	Colluders	122
4.4.6	Author Identification	123
4.4.7	Encryption	123
4.5	Summary	123
5	Design & Simulation	125
5.1	System Design	125
5.1.1	Nodes (Physical Devices)	125
5.1.2	Users	126
5.1.2.1	Local Priority Queue	127
5.1.3	Protocol Messages	127
5.1.3.1	Questions	127
5.1.3.2	Answers	127
5.1.3.3	Feedback	128
5.1.3.4	Joins and Neighbour Management	129
5.2	Performance Metrics	129
5.2.1	Answer Quality	129
5.2.2	Attention Consumed	130
5.2.3	Percentage of Unanswered Questions	130
5.2.4	Path Lengths	130
5.3	Simulation	131
5.3.1	PlanetSim	131
5.3.1.1	Layered Approach	131
5.3.1.2	Node and Node Handles	132

5.3.1.3	Message Queues	132
5.3.1.4	Simulation Steps and Lengths	132
5.3.1.5	Route Messages	134
5.3.1.6	Behaviours	134
5.3.2	Networking Topology	134
5.3.3	Node Structure	135
5.3.4	Question Generation	136
5.3.5	Answer Generation	136
5.3.6	Feedback Generation	137
5.3.7	User States	137
5.3.8	Network Churn	137
5.3.9	Random Number Generators and Seeding	139
5.4	Experimentation	140
5.4.1	Data Distributions	140
5.4.2	Configuration	141
5.4.3	Network Set up and Creation	141
5.4.4	Gathering Results	142
5.4.5	Processing Results	143
5.4.6	Experimentation Debugging	144
5.4.6.1	Step Through	144
5.4.6.2	Network Visualisation	144
5.5	Summary	145
6	Evaluation	147
6.1	Network Size and Simulation Length	147
6.1.1	Running Times	148
6.1.2	Memory Usage	149
6.1.3	Experimental Defaults	149
6.2	Pheromones	151
6.2.1	Constant Increase Versus Proportional	152
6.2.2	Warm up periods	152
6.2.3	Pheromone Rate, Feedback and Attention	154
6.2.4	Pheromone Maximums and Default levels	154
6.2.5	Balance of Quality and Attention	158
6.2.6	Load Balancing and Initial Values	158
6.2.7	Pheromone Evaporation	162

6.2.8	Adjustment of parameter choices	162
6.3	Generic Protocol	163
6.3.1	Network Properties	163
6.3.2	Question Time-To-Live Values	163
6.3.3	Number of Answers Required	165
6.4	Summary of the Generic Protocol	168
6.5	User Model	170
6.5.1	Transitional Probabilities	170
6.5.2	Priority Queue Size	172
6.5.3	Question-Asking Rate	172
6.5.4	Reading and Writing Abilities	174
6.6	Summary of User Model Evaluation	174
6.7	Network Churn	176
6.7.1	Answer Quality	176
6.7.2	Attention	179
6.8	Scalability	179
6.9	Results	182
6.9.1	Best Answer Quality	182
6.9.2	Average Answer Quality	183
6.9.3	Path Lengths and Network Load	183
6.9.4	Unanswered Questions	185
6.9.5	Consumed User Attention	185
6.9.5.1	User Attention Per Question	185
6.9.5.2	Total Attention Per User	186
6.9.6	Exponentially Mean Weighted Averages	186
6.9.7	Summary	189
6.10	Attack Models	189
6.10.1	Eager Answer	189
6.10.2	False Feedback	189
6.10.3	Question Blocker	191
6.10.4	Answer Blocker	191
6.10.5	Feedback Blocker	191
7	Conclusions & Future Work	194
7.1	General Observations & Lessons Learned	195
7.2	Future Work	195

7.2.1	Broken Routes	196
7.2.2	Incentives	196
7.2.3	Real World Implementation	196
7.2.4	Routing Variations	196
8	Appendix	198
	Bibliography	214

List of Figures

2.1	A small example MANET.	24
2.2	A small example social network.	27
2.3	Twitter mood map.	28
2.4	Yahoo! Answers: question and answering process.	34
2.5	Onion routing map.	48
2.6	Binary bridge experiment.	53
2.7	Short branch experiment.	54
2.8	British Telecom synchronous digital hierarchy (SDH).	57
3.1	SQL database structure of Yahoo! Answers corpus.	73
3.2	Yahoo! Answers questions per month in 2006.	75
3.3	Corpus Q&A counts.	76
3.4	Number of answers per question.	77
3.5	Question category popularity.	78
3.6	Answer category popularity.	79
3.7	Category popularity within the Yahoo! Answers dataset.	80
3.8	Number of interest and expertise categories per user.	81
3.9	Number of best answer distribution.	81
3.10	Q&A length distribution.	82
3.11	Bespoke crawler setup.	84
3.12	Twitter users timezones.	86
3.13	Twitter users around ‘Brighton’.	87
3.14	Twitter users around ‘London’.	87
3.15	Distribution of days users are active.	88
3.16	Distribution of tweets per user.	89
3.17	Inter-tweet time distribution.	89
3.18	Inter-tweet reply time distribution.	90

3.19	Inter-tweet question time distribution.	91
3.20	Number of replies to a tweet distribution.	92
3.21	Conversation length distribution.	93
3.22	Markovian user state model.	95
3.23	Possible churn scenarios.	100
3.24	Possible churn scenario PDFs.	101
4.1	Flooding: send question to all links.	104
4.2	Random Hops: uniform path selection.	105
4.3	Protocol question message sequences.	107
4.4	Protocol answer message sequences.	107
4.5	Protocol feedback message sequences.	108
4.6	Local category pheromone values.	110
4.7	Local category pheromone values.	111
4.8	Probability of path selection.	114
4.9	Stigmergic V1: path selection with scent levels.	115
4.10	Skipping network users to reach experts.	116
4.11	Stigmergic V2 & V3: loopback routing table entry.	118
5.1	An example of a small Q&A network.	126
5.2	User with node and queues.	133
5.3	High level abstracted node structure.	135
5.4	High level abstracted message structure.	136
5.5	Churn: Inter-arrival time and session duration.	138
5.6	java.util.Random value distribution.	140
5.7	Initial network members.	142
5.8	35 Node visual with pheromone scent levels.	145
5.9	100 Node visual with pheromone scent levels.	146
6.1	Flooding simulation times.	148
6.2	Random simulation times.	149
6.3	Stigmergic simulation times.	149
6.4	Flooding memory usage.	150
6.5	Random memory usage.	150
6.6	Stigmergic memory usage.	150
6.7	Best answer quality against pheromone update values (constant). .	152
6.8	Best answer quality against pheromone update values (proportional).153	

6.9	Exponentially weighted mean average answer quality.	154
6.10	Best answer quality against pheromone update values (constant). . .	155
6.11	User attention against pheromone update values (constant).	155
6.12	Best answer quality against pheromone update values (proportional).156	
6.13	User attention against pheromone update values (proportional). . .	156
6.14	Default and maximum value effects on answer quality.	157
6.15	Default and maximum value effects on user attention.	157
6.16	Ratio between quality and attention.	158
6.17	V2 pheromone startup value against answer quality.	159
6.18	V2 pheromone startup value against bombardments.	159
6.19	V2 pheromone startup value against unanswered questions.	160
6.20	V3 pheromone defaults against answer quality.	160
6.21	V3 pheromone defaults against unanswered questions.	161
6.22	V3 pheromone defaults against overloads.	161
6.23	Pheromone evaporation against answer quality.	162
6.24	Pheromone evaporation against user attention.	163
6.25	Network visualisation of 1000 node random network.	164
6.26	(Stigmergic) TTL values against unanswered questions.	165
6.27	(Stigmergic) TTL values against answer quality.	166
6.28	(Stigmergic) TTL values against user attention.	166
6.29	(Random) TTL values against unanswered questions.	167
6.30	(Random) TTL values against answer quality.	167
6.31	(Random) TTL values against user attention.	168
6.32	Number of answers required consequences.	169
6.33	User model effects on best answer quality and user attention. . . .	171
6.34	Queue sizes causing overloading.	172
6.35	Question asking rate consequences	173
6.36	Reading and writing abilities against mean best answer quality. . .	174
6.37	Reading and writing abilities against unanswered question.	175
6.38	Reading and writing abilities against user attention.	175
6.39	Exponentially mean weighted average (C1).	177
6.40	Exponentially mean weighted average (C1).	177
6.41	Exponentially mean weighted average (C2).	178
6.42	Churn effects on answer quality.	178
6.43	Churn effect on user attention.	179
6.44	Best answer quality.	180

6.45	Average answer quality.	180
6.46	Attention per question.	181
6.47	EWMA quality.	181
6.48	Simulation parameters.	182
6.49	Best answer quality results.	183
6.50	Average answer quality results.	184
6.51	Network load results.	184
6.52	Unanswered questions results.	185
6.53	Attention per question results.	186
6.54	Attention per user results.	187
6.55	Exponentially mean weighted average (C0).	187
6.56	Exponentially mean weighted average (C1).	188
6.57	Exponentially mean weighted average (C2).	188
6.58	Eager answerer effect on quality.	190
6.59	False feedback effect on quality.	190
6.60	Question blocking effect on quality.	191
6.61	Answer blocking effect on quality.	192
6.62	Feedback blocking effect on quality.	192
8.1	Yahoo! Answers Categories 1 to 10.	203
8.2	Yahoo! Answers Categories 11 to 20.	204
8.3	Yahoo! Answers Categories 21 to 27.	205
8.4	AA1 quality.	206
8.5	AA1 attention.	206
8.6	AA1 ratio.	206
8.7	Low Question Rate and Attention (AA1).	206
8.8	AA2 quality.	207
8.9	AA2 attention.	207
8.10	AA2 ratio.	207
8.11	High Question Rate with Low Attention (AA2).	207
8.12	BB1 quality.	208
8.13	BB1 attention.	208
8.14	BB1 ratio.	208
8.15	Low Question Rate with High Attention (BB1).	208
8.16	BB2 quality.	209
8.17	BB2 attention.	209

8.18 BB2 ratio.	209
8.19 High Question Rate with High Attention (BB2).	209
8.20 CC1 quality.	210
8.21 CC1 attention.	210
8.22 CC1 ratio.	210
8.23 Low Question Rate and Attention with Churn (CC1).	210
8.24 CC2 quality.	211
8.25 CC2 attention.	211
8.26 CC2 ratio.	211
8.27 Low Question Rate and Attention with Churn (CC2).	211
8.28 DD1 quality.	212
8.29 DD1 attention.	212
8.30 DD1 ratio.	212
8.31 Low Question Rate with High Attention and Churn (DD1).	212
8.32 DD2 quality.	213
8.33 DD2 attention.	213
8.34 DD2 ratio.	213
8.35 High Question Rate with High Attention and Churn (DD2).	213

List of Tables

2.1	Q&A service comparison.	43
2.2	Stigmergic comparison.	61
3.1	Questions asked per year.	74
3.2	Average WPM for text entry methods.	97
5.1	Question message structure.	128
5.2	Answer message structure.	128
5.3	Feedback message structure.	129
6.1	Initial network characteristics.	164
8.1	Investigation details.	202

Introduction

This thesis provides an investigation into a plausibly deniable yet fair Question and Answering (Q&A) service over ad hoc networks. Many research areas including deniable routing, expert retrieval, peer-to-peer networking and user modelling are drawn together to address this challenge.

Online Q&A services allow users to appeal for answers to questions from a very large audience, and the larger the collection of users to ask, the greater potential to find expertise on a wide range of subjects. Recently, Q&A services which make use of social networks and real identities have begun to appear [1, 2]. The accountability from identities being linked to questions and answers is believed to increase both trust and answer quality [1]. However, these approaches may cause users to only ask specific types of questions, presenting a limited and restricted service in terms of utility. For example, users may wish to avoid asking questions of a sensitive nature to a vast unknown audience or may be reluctant to participate in exchanges where there is a chance of being interpreted incorrectly [3].

Within a distributed environment it is possible to hide among the networked crowd of individuals [4, 5, 6]. Such crowd-based techniques are suitable for creating a plausibly deniable Q&A network infrastructure, in order to facilitate the anonymous exchange of questions and answers. Ant-inspired routing or stigmergy is particularly applicable to the task of deniable Q&A as it is not identity-based, allowing paths to emerge in the network without specific users being explicitly addressed. Privacy is known as an emergent benefit of stigmergy however, it is rarely the focus of research in this area [7, 8, 9, 10, 11, 12, 13, 14].

Human expertise and user attention are precious resources within Q&A networks. Unnecessarily consuming the time and effort of users is undesirable

and should be avoided. It is important to consider the implications caused by constantly bombarding the most knowledgeable members of the network. Routing should aim to be fair, spreading the attention costs throughout the network where possible.

The field of Expert Retrieval (ER) aims to locate a suitable user to answer a specific query. Existing approaches to this task do not take into account the privacy of individuals, the levels of attention consumed or the fair routing of data [15, 16, 17, 18, 19, 20, 21]. This thesis investigates a distributed approach towards these largely ignored issues.

With the introduction of the fourth generation of mobile Internet (4G), it seems inevitable that users will be continuously connected throughout the day via mobile devices. This assertion motivates and supports further research towards human-orientated services such as Q&A which address real user concerns and needs at this time when digital communication is more prominent than ever.

A deniable question and answer service may promote communication on sensitive topics such as health, religion and professional or personal issues. A user may feel more comfortable discussing certain topics without disclosing their identity. Possible scenarios of such a system are provided, these example situations would motivate the need for our system. Together they also motivate the need for a general system, rather than one which focuses on a particular topic.

- Questioning specific policies or actions of the controlling government.
- Questions where a lack of knowledge might be embarrassing.
- Linking between answers and ownership. A user may prefer to be allowed to say “don’t ask me how I know this”.
- Sexual and relationship issues.
- The discussion of an embarrassing or particularly unpleasant physical or psychological ailment.
- Questions surrounding specific religious teachings or practices.
- Requesting for help with professional difficulties, such as how to correctly solve an issue.

1.1 Contribution

This thesis presents a new and non-trivial research contribution for locating human expertise in a distributed environment, both deniably and fairly. The task of Q&A is complex, due to the shortage of human expertise and the possible routes found within the network. This issue is complicated further by the need for plausible deniability for authors and taking into consideration the levels of human attention used to generate adequate answers. In a distributed environment, questions need to be routed towards the members of the network who stand the best chance of answering, while simultaneously avoiding bombardment and wasted user effort.

In this thesis a distributed service is motivated and designed, and possible question-routing approaches are compared and evaluated. An ant-inspired protocol is presented as an effective means to deniably and fairly locate expertise. To the best of the authors knowledge, this is the first time stigmergy has been used for this purpose.

A large Q&A dataset is used to create a user model for the realistic representation of typical Q&A users with regard to a range of features, such as topics of interests and expertise levels.

The application of the proposed protocol is compared experimentally against naïve approaches via extensive peer-to-peer network simulations. Malicious attacks against the routing protocol are also described and evaluated, exposing potential strengths and weaknesses.

1.2 Thesis Structure

A brief summary of each thesis chapter follows:

2 – Literature Review

This chapter draws together supporting material which fuels the aims and contributions of the thesis. It surveys relevant research on Q&A services, ER, digital security and privacy, swarm intelligence and peer-to-peer networking.

3 – User Modelling

An analysis of the Yahoo! Answers community and their questions & answers is coupled together with an analysis of data collected from Twitter

during the course of this study. The aim of this chapter is to uncover and identify suitable user traits and data distributions to aid Q&A user modelling and thus the simulation of realistic Q&A networks.

4 – Deniable Routing for Q&A

This chapter presents several naïve approaches and an ant-inspired routing protocol to deniable question routing as solutions to the research challenge. The ant-inspired approach aims to encourage questions to flow towards the members of the network who stand a better chance of answering well while adhering to the privacy requirements by hiding requests within the networked crowd of individuals.

5 – Design and Simulation

This chapter presents Q&A system entities and procedures required to simulate Q&A networks. Using the user model and routing approaches along with the techniques and principles in this chapter, a platform for the simulation of a distributed Q&A network is presented. Key performance metrics are defined to allow for comparison to be made in the evaluation chapter.

6 – Evaluation

An in-depth study of the routing approaches in a simulated environment is presented in this chapter. The approaches are analysed and compared to uncover their strengths and weaknesses. The quality of generated questions, the proportion of unanswered questions, the effects of churn on routing, user attention transition probabilities, question frequencies, pheromone update rules, network sizes and scalability are all considered. The evaluation ends with a collection of results using a set of suitable and realistic choices for simulation variables and investigations of possible attack models.

7 – Conclusion and Future Work

Finally conclusions are drawn and future work is described in relation to the evaluation and lessons learnt during the research period.

1.3 Published Work

Elements of this thesis have been selected for publication or presented in the following works:

1. **Fleming S.**, Chalmers D., and Wakeman I. *A Deniable and Efficient Question & Answer Service Over Ad Hoc Social Networks*, Special Issue of Information Retrieval (SI). Springer 2012.
2. Basu, A., **Fleming, S.**, Stanier J., Naicken, S., Wakeman, I. and Gurbani, V. K. *A Survey of Peer-to-Peer Network Simulators and Simulations*. Submitted to ACM Computing Surveys.
3. Chalmers D., **Fleming S.**, Wakeman I., Watson D. *Rhythms in Twitter*. International Workshop on Social Object Networks (SocialObjects) IEEE 2011
4. **Fleming S.**, Chalmers D., Wakeman I. *Routing in Question and Answer Networks*. Presentation in the proceedings of Multi-Service Networks (MSN10), Oxford, UK, 2010.
5. **Fleming S.**, Chalmers D., Wakeman I. *Can We Exploit the Wisdom of Large Ad Hoc Crowds?* Poster in the proceedings of Workshop on the Future of Social Networking: Experts from Industry and Academia (SOCIALNETS), Cambridge, UK, 2010.

Literature Review

Human reliance on computers continues to develop. As well as processing, storing and manipulating data extremely efficiently computers are increasingly being used to facilitate communication. This allows distributed systems and applications to co-ordinate and empowers owners by enabling communications via an ever-increasing collection of devices. This developing technology has had a huge impact on how and why we choose to communicate as our understanding of digital society grows. As computer-mediated communications are still a somewhat recent phenomenon and are constantly evolving, this is an active and interesting research area which warrants further investigation and discussion.

2.1 The Internet and the World Wide Web

The Internet (*Internetwork*) can now be thought of as a *ubiquitous communication medium*. It can be instantly accessed via many modern devices including desktop computers, mobile phones and portable hand-held computing devices such as tablet computers [22] – an individuals access to multiple devices was identified by Weiser [23] in his vision of the 21st century.

The wealth of knowledge available on the Internet is enormous and is growing rapidly [24]. Information, services and people can all be found within the realms of the Internet via the World Wide Web (WWW). The WWW was originally a means for viewing documents online, supporting structures which allow links to be made to other related documents known as web pages. These web pages were structured using the HyperText Markup Language (HTML) [25], which allows documents to be created and published simply, while allowing web browsers to render the documents to provide content and structure. The WWW today promotes strict structure rules for HTML documents and

also supports a wide range of additional document encodings (such as XML [26]), as well as multimedia features such as video and radio streaming.

2.1.1 Mobile ad hoc Networks

Mobile ad hoc Networks (MANETs) are a fascinating prospect, presenting us with many exciting and interesting possibilities, paving the way for the next generation of network applications and architectures. With the increasing ubiquity of powerful portable and hand-held devices such as notebooks and smartphones, all of which have local interconnection possibilities through wireless, Bluetooth or infrared technologies, we now have the ability to form networks at any location.

MANETs are networks which consist of mobile devices that are free to move around independently of one another, working together to route data around the network. As a particular device roams around an environment it will lose and gain connections within the network – this dynamic feature of networking is an interesting research challenge. An example MANET of the various links between network entities can be seen in Figure 2.1.

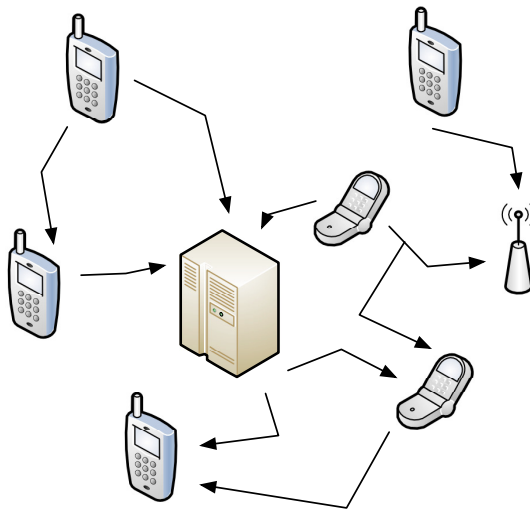


Figure 2.1: A small example MANET.

The potential for emergent groups and crowds of individuals to connect with one another to form ad hoc services facilitating sharing and collaboration

is a realistic expectation of the future. As Moore's law¹ continues to unfold, the resources available to the devices which form ad hoc networks will increase and develop. Similarly, as computer devices become cheaper and smaller, an increasing number of mobile devices should exhibit a wider range of communication tools to facilitate the creation of ad hoc networks.

The size, extent and use of MANETs is perhaps currently unclear – however, MANETs have great potential, for example, the formation of such networks would work extremely well for specific events like festivals, large markets and conferences, and within communities and groups. The size of a MANET is determined by the number of individuals present or interested who have such a device. Taking into account the current estimated proportion of people who have a mobile phone and a recent prediction by Gartner² that by 2013 access to the Internet via mobile devices will exceed access via desktop computers – we may anticipate that most individuals in the not so distant future will have a MANET capable device and could participate in ad hoc networks.

2.1.1.1 Internet-Based Mobile ad hoc Networks

MANETs are also used to form networks which connect with an Internet gateway in the form of an Internet Based Mobile ad hoc Network (iMANET), providing Internet connectivity to the network members.

A recent example of the novel use of a MANET is the improvement of traffic information systems [27]. Vehicular ad hoc networks (VANETs) are formed by routing requests through a MANET to provide privacy by hiding the source of a particular request. Unfortunately, these networks require the voluntary participation of other devices in the local vicinity. One alternative possibility would be to replace VANETS via a decentralised Peer-to-Peer (P2P) overlay network across the Internet in order to expand and extend the potential group from which to form ad hoc connections. Expanding local ad hoc networks to global P2P networks could increase the possibilities and usefulness of such applications.

Rybicki et al. [27] recognise the clear privacy issues associated with such a VANET and the trade off between trustworthiness and the potential tracking of user movements. The authors suggest that improved privacy over a centralised

¹Gordon Moore's observation in 1965 that the number of transistors per integrated chip increases exponentially with time.

²<http://www.gartner.com/it/page.jsp?id=1278413>

system is achieved by distributing sensitive movement data over many different nodes, in such a way that no one node knows the entire history of a given node. Although the authors do not elaborate on the method of achieving such decentralisation to improve user privacy, it is clearly an important and interesting research challenge.

2.1.2 The Growth of User Generated Content

The term Web 2.0 is used to describe websites which allow users to generate content and to interact and collaborate with one another. Web 2.0 can be loosely broken down into several common structures, including but not limited to: social networking, blogging, micro-blogging and resource sharing. Examples of successful systems based on Web 2.0 technologies include popular services like Facebook³, Myspace⁴, Twitter⁵ and Flickr⁶.

Web 2.0 provides facilities for users to communicate such as Asynchronous JavaScript and XML (Ajax) powered chat features and direct messaging similar to electronic mail (e-mail). Unlike traditional e-mail, user communication facilities are provided and managed by the Web 2.0 providers rather than an Internet Service Provider (ISP) or industry mail server. Users are found and contacted via an online profile related to a distinct identity, rather than an e-mail address alias at a particular domain.

Web 2.0 users are empowered by the ability to set up an online identity, typically via a personal profile or feed page. This profile ‘hub’ provides a central location acting as a virtual public address. Communications and actions are typically public, such that activities can be viewed in a timeline by other users. As many Web 2.0 sites provide layouts and features specifically for the mobile phone market, they are often considered to be mobile applications.

Web 2.0 technologies create vast quantities of user generated content, encourage social communities and facilitate communication. For example, Facebook provides a means for friends in keep in contact with one another easily and on demand. An example social network can be seen in Figure 2.2, showing users and the links between them.

³<http://www.facebook.com>

⁴<http://www.myspace.com>

⁵<http://www.twitter.com>

⁶<http://www.flickr.com>

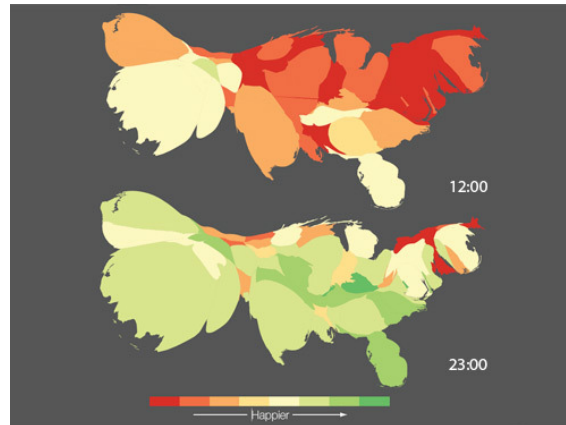


Figure 2.3: Twitter mood map.

Web 2.0 technologies are particularly interesting from a research perspective, as they allow observations and conclusions to be made with regard to computer-mediated social interactions and the behaviour of Web 2.0 users. In the past, user interactions and behaviours made via computers have been somewhat hidden from the general public and research community. Only those organisations and institutions responsible for the communications would have had direct access to this information. Web 2.0 technologies allow access to a vast amount of information, which can be analysed to find implicit information; for example work undertaken here at the University of Sussex in the Foundations of Software Systems group by Feizy et al. [30, 31].

The collection of these Web 2.0 services for a specific user could be used to help define a digital footprint. A digital footprint can be considered as the set of all digital activities associated with a particular individual or organisation. Due to the extent of information stored on such technologies, users may wish to control which elements are archived or logged to allow for “the right to be forgotten”⁸.

While there are many advantages to Web 2.0 technologies, for the user they can have a detrimental effect, as can many instant computer-mediated communication technologies. They are shaping the way we communicate and make personal information and opinions available to a much wider audience. For example, it is now much easier to be held accountable for past activities and opinions. Users may make a statement they later regret, but once it’s

⁸<http://metro.co.uk/tech/858339-facebook--court-action-over-private-data>

been published to their online community there are limited opportunities to retract it. There have been stories in the press describing employee dismissals due to comments made over social networking services⁹. Obviously this is a relatively new phenomenon, because in the past comments were more likely to be made during face-to-face meetings, and the ability to publish something online while in a state of anger, upset or inebriation is all too easy. There have also been cases of dismissals after company employees have had inappropriate social networking discussions in which customers have been insulted¹⁰. Users must be careful to control their updates, or indeed who has access to them.

2.1.3 Knowledge Markets and Public Search

We are currently in an age where there is a race to control datasets [32]. More and more organisations are valued on the quality of the data they possess rather than the profits they have made. For example, Facebook was valued at \$50 billion in 2011, despite only having \$2 billion in sales¹¹.

A description of a Web 2.0 site called Quora (an online Q&A service) states that:

“People use Quora to document the world around them. Over time, the database of knowledge should grow and grow until almost everything that anyone wants to know is available in the system. When knowledge is put into Quora, it is there forever to be shared with anyone in the future who is interested.”¹²

While many organisations are now aiming to acquire data regarding user generated content, it is also inherently searchable via search portal sites. This is useful to help users locate information but may be detrimental to users who publish but later lose control of what happens with the data.

2.1.4 Information Overload

In our modern world, we may be contacted via many different methods: telephone, text message, multiple e-mail addresses, a huge range of possible instant message applications and Web 2.0 applications such as Facebook and Twitter.

⁹<http://www.dailymail.co.uk/news/article-1206491/>

¹⁰<http://news.bbc.co.uk/1/hi/7703129.stm>

¹¹<http://www.wired.com/epicenter/2011/01/facebook-valuation/>

¹²<http://www.quora.com/about>

We need to be able to manage our incoming communications without being overloaded by the enormity of the stream.

Even back in the infancy of the modern web, Palme [33] saw the problems facing users when empowerment is given to those users who wish to initiate communications – rather than those at the receiver’s end. The worst case scenario is that important or interesting communications may be lost within the large volume of communications received. However, it is clear that computer mediated communications do scale with the number of users present, unlike traditional face-to-face meetings which can become less effective as numbers increase. It is essential for large groups to be able to collaborate effectively.

Microsoft have investigated the cost of interruption [34], stating that interruptions at the wrong time can be costly to business. Therefore it is of paramount importance to wisely mediate the flow of alerts and communications to users. Users who participate and help others in networked applications should be treated as a highly valuable resource by the network service, and in turn treated with respect. The users should not be bombarded or abused as a boundless resource. Several distinct *states* of attention within the office setting have been identified, including: *high-focus solo activity*, *low-focus solo activity*, *conversation in office*, *presentation*, *meeting*, *driving*, *private personal time*, *sleeping* and *now available*, all of which refer specifically to a description of a user’s attention, focus or workload. The cost of interruption during these different states of attention will be of varying consequence. The authors aim to mediate communications while minimising interruption costs. Three naïve levels (low, medium and high) of interruption cost are defined, in order to aid the classification and prediction of the true cost. Unfortunately, the authors found that one model does not fit all users, and applying a personalised model from one user to the next may yield poor prediction performance.

Overall, Horvitz and Apacible [34] were able to develop personalised predictive attention models from detailed user activity monitoring over a five-hour period. The work relies on comprehensive user details (including face cameras, monitoring all computer activities, calendar and acoustic sensor data) to build the models and as such is a rather invasive approach. It does however, promote the idea that certain times are better than others for communication.

2.2 Question and Answer Services

It is now possible to appeal for help and advice with any question via a number of means. Perhaps the simplest is to ask friends, colleagues and acquaintances directly via face-to-face encounters or direct targeted communications such as telephone, text message (SMS), e-mail or letter. This system works well if a social network contains a varied number of people or sources. However, it is becoming an increasingly attractive option to appeal to a much wider audience for knowledge generation and exchange via electronic resources such as Q&A services [35]. Turning to an electronic Q&A service may be beneficial when traditional search engines fail to produce useful results. The Internet provides a wealth of data but finding a specific contextualised piece of information can prove to be time consuming or problematic.

In general, users may provide answers to questions for a number of reasons [36], including altruism (selfless concern for the welfare of others), egoism, and self-rewards such as the desire to learn, develop and expand knowledge, or to demonstrate proficiency in a given subject area. Services such as Mechanical Turk¹³ are also popular for benefiting from the wisdom of the crowd via financial incentives.

As well as online Q&A services, text-message based pay for Q&A services such as Ask Ollie¹⁴ or 6336¹⁵ exist, whereby you can ask any question via text message at any time, and receive an answer within a short period of time. This form of Q&A is popular because it is extremely convenient; however it does incur a cost. Unlike Q&A services, there is no option to query the identity of the answerer, meaning that the user must place a certain amount of trust in the service. The user does not know who the answerer is, how qualified they are to answer the question, whether they are biased at all, whether they are being paid to provide the service and so on.

After an initial burst of popularity, perhaps partially due to the novelty effect of such a service and the fact that mobile internet was not particularly common, text-message based services are no longer prevalent. The majority of mobile and hand-held devices are now Internet capable, so it makes sense for users to take advantage of the free services on offer via the Internet and ask questions to a wider range of users.

¹³<http://www.mturk.com>

¹⁴<http://www.askollie.com>

¹⁵<http://www.aqa.63336.com>

Questions can be defined as being *factual, opinion, advice or spam* [36] and Q&A services involving humans, as opposed to automatically-generated answers, are thought to be particularly good for gaining useful answers to opinion- and advice-based questions. Hsieh et al. [36] analysed 800 random questions posted by users via Mahalo Answers¹⁶ and used crowd-sourcing to identify question types. The question analysis highlighted that users did not ask questions as they felt reluctant to reveal a certain lack of knowledge, believing that it may have a negative impact on their external reputation. However, these effects can be mitigated to a degree by the fact that questions are not specifically targeted to a particular answerer and that reputation are typically linked to a pseudonym rather than a users obvious real identity.

Other work, such as [35, 36], investigates whether financial incentives support question answering, specifically in terms of quality and archival value. The authors dub these ‘pay-for-answer Q&A services’. The difficulty and nature of questions is often used to determine if a user is willing to pay for an answer, and how much monetary value they place on said information. Research shows that the monetary value of a question has an effect on the answers – they are more likely to receive an answer but this does not necessarily have an affect on the answer quality. It would seem that those with the knowledge to answer user questions are often willing to share their expertise for free.

Other work by Harper et al. [37] has investigated ‘informational’ rather than ‘conversational’ questions for archival qualities using text classification techniques to distinguish between them. The authors discuss how categories and text parsing can help to identify the class of questions. For example “bag of words” which classifies questions by the types of words used such as “I” and “You”, as well as terms like “Where” and “How”. Phrases are also considered to be a strong predictor, such as “is there”, “do you” and “would you”. The authors hope that using such techniques can help to direct questions to the appropriate place. Techniques involving *how* to target specific questions towards particular users who may be able to provide an appropriate answer is a extremely useful and helpful element to Q&A which appears to have received little direct attention from the research community.

Various Q&A services were examined during the course of this thesis, a comparison of which is presented in Table 2.1, which shows the means of accessing the services, the routing, feedback support and privacy considerations.

¹⁶<http://www.mahalo.com/answers/>

2.2.1 Yahoo! Answers

One of the most established active online Q&A services in use today is Yahoo! Answers¹⁷. It has tens of millions of users in the United States alone, where the service originated, and almost one hundred million users worldwide. Yahoo! states that “*Everyone has life experience and knowledge about something, and Yahoo! Answers provides a way for people to share their experience and insight.*”¹⁸. Ultimately, if you are able to tap into the knowledge of those who have expertise in a subject it is likely that you can gain useful and correct information both quickly and efficiently, provided you know how to ask for it. As Q&A services typically follow the processes of Yahoo! Answers for question asking and answering, below is an overview of the question and answer process seen in Yahoo! Answers.

Question Asking:

- New questions are assigned to a question category to help others find them.
- Questions remain open for 4 days as default, this period may be extended or reduced.
- After the question has been answered, a best answer may be selected after 1 hour. Alternatively the community may choose the best answer.

Question Answering:

- Questions may be located via category listings or search.
- An answer may be given to any open question.
- Useful answers earn users points which can be used to ask additional questions and provide a ranking system.

Figure 2.4 presents a flow diagram representing the question and answering process on Yahoo! Answers¹⁹.

One piece of work describes a study which took place during an internship at Yahoo! Incorporated to investigate Yahoo! Answers data from August 2005-2006 [38]. This work uses an algorithmic approach to identify potential experts

¹⁷<http://answers.yahoo.com/>

¹⁸<http://help.yahoo.com/l/us/yahoo/answers/overview/overview-55778.html>

¹⁹http://uk.answers.yahoo.com/info/product_tour

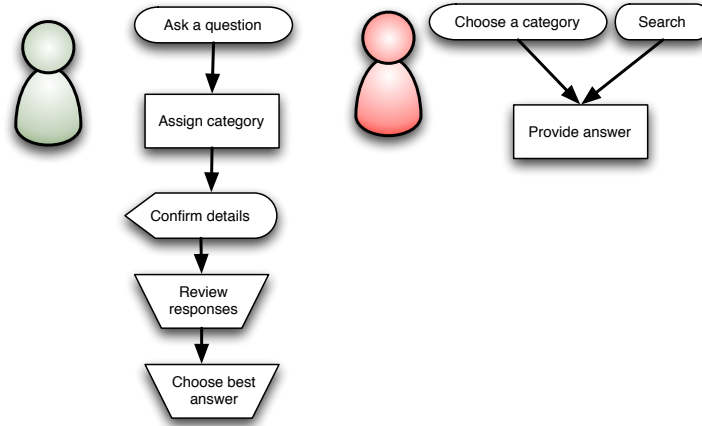


Figure 2.4: Yahoo! Answers: question and answering process.

within the dataset. A *Hub score* is assigned to users, indicating those users that have asked many interest-generating questions (those questions which receive many answers). In addition, an *Authority Score* is used to determine whether a user is knowledgeable by having many best answers to interest-generating questions. The key message here is that best answers are more powerful or representative of author expertise when assigned in a competitive answer group consisting of many answers. However, the authors argue that best answers are of questionable value, as the votes are subjective in nature. Users may incorrectly identify the best answer among a set and in addition it may be difficult to differentiate between varying levels of quality. Furthermore, answers may be branded as the best maliciously. This work also states that answerers are not necessarily domain experts focusing on a single topic, they often have diverse interests and are eager to explore various question types, which begs the question how knowledgeable they really are about each of these types. In addition, the authors state that the users with the highest number of best answers are not necessarily the most authoritative – perhaps this is an artefact of the popularity of some categories within the system (it may be easier to be nominated for providing a best answer in popular categories with lots of questions).

It is noted that subjective responses have no good or bad answers, however, it could be argued that they do in fact have better answers depending on the expertise and experience of the answerer.

Most interestingly, the study indicates that the more focused questions may not receive adequate answers, “*possibly because they never get exposed to a competent answerer (no such user exists in the system, or the question does not get routed to him/her)*” [38]. As Yahoo Answers! does not adequately support discussion or true real time interactions, it is suggested that personalisation mechanisms are used to help route interesting questions to potential answerers. Users could be notified about relevant questions to help minimise the percentage of questions that are poorly answered or not answered at all. **It is clear that there is a need for new research into how to help target specific questions towards specific users with the appropriate knowledge for answering within Q&A services.**

Further research has been conducted to investigate why some questions do not receive answers from online Q&A communities [3]. It is assumed that users will not answer questions if they are uninterested in the subject or if they are unable to provide an answer – but these assumptions do not provide a complete picture. Top-rated and regular contributors appear to not answer questions for similar reasons, for example, questions which already have a high number of responses are likely to be avoided because their contribution may be “*lost in the crowd and not be read*” [3]. Most importantly, the respondent’s perception of how the asker will *receive, interpret and react* to their answers is paramount. Users do not wish to get reported for abuse and risk losing access to their online community. This ties in with concerns over the interpretation of answers. For example, a quote from one interviewee in this study states “*Certain questions I don’t reply to because I am afraid that if I express my personal opinion, I might offend someone*” [3]. Respondents are wary of answering on certain topics when there is a chance that the response may be misinterpreted or misconstrued.

In summary, Yahoo! Answers was one of the original Q&A sites, and is still extremely popular and useful today. Research focusing on Yahoo! Answers shows that there are several issues and concerns surrounding how a user may be perceived based on their interactions with the service. In addition, how it is possible to promote the answering of questions is also of interest. Questions need to be presented to those members of the community who are capable of answering and this is a key requirement which is investigated in this thesis.

2.2.2 Aardvark

During the creation of this thesis, Aardvark²⁰ emerged as a new and innovative Q&A service aiming to find answers to questions in real time, using routing via instant messaging and e-mail. Aardvark is a social search engine which utilises a users' social network of users who questions may be routed to. Aardvark attempts to find "*the right person to satisfy a user's information need*" [1]. It suggests that research questions open up in information retrieval in regard to *human expertise classification*, *implicit network construction* and *conversational design*. The questions generated via Aardvark appear to be long, highly contextualized and subjective – which the authors suggest are the types of queries that are not well-served by traditional search engines. Users on the system are surprisingly active, both in asking and answering. The authors state that Aardvark performs very well on queries which deal with *opinions*, *advice*, *experience* and *recommendations*. However, asking for advice within your social network may not always be appropriate or desirable, particularly with regard to sensitive topics like health and religion.

The Aardvark system allows users to manually indicate at the sign up stage which topics they have expertise in, in order to bootstrap the system. Additionally, users may indicate which topics they trust a given user's opinions about.

The topics associated with each user are recorded in a *forward index*, which stores all users as well as a scored list of topics and additional information about a user's behaviour. There is also an *inverted index* which stores topic IDs and a sorted list of users who have expertise in that topic. The inverted index also stores scored lists of users for features such as *answer quality* and *response times*. Aardvark has the ability to extract information from structured sources such as Facebook or Twitter, although it can be argued that this information could be wrong or unhelpful at times, for example, information posted from another user on a given profile page.

Aardvark uses a *Question Analyzer* to "determine the appropriate classifications and topics for the question" [1]. This is an essential feature of the application which determines the context of a question and who it should be sent to. Automatic question classification can have its problems, for example, assigning a question to an inappropriate category due to user error (selecting

²⁰<http://www.vark.com>

an incorrect category) or lack of knowledge about a given subject within the analyzer.

Classifiers are presented as part of the question analyzer – specifically a **NonQuestionClassifier** which prevents invalid questions from entering the system. Furthermore, an **InappropriateQuestionClassifier** “*determines if the input is obscene, commercial spam, or otherwise inappropriate content for a public question-answering community*”, thus helping to prevent abuse of the system, something which the designers have identified as a serious issue.

Aardvark uses a simple question asking loop, where users are identified and asked sequentially in turn, until a required number of answers (or answer offers) are gained. It also allows for conversations to take place between users following an initial interaction in the form of a question and answer.

Aardvark’s model attempts to assign a probability that a specific user will successfully answer a given question. Aardvark calculates a *query-dependent* score (matching a user with question topics) and also a *query-independent* score (based on profile matching and social connectedness).

Aardvark employs an interesting feature intended to help improve confidence in question routing. If a user and their friends have expertise in a given topic they are considered to be more of an expert in the subject area than a user without any matching expert friends. This could be thought of as a similar mechanism to the Hub and Authority score discussed earlier.

Aardvark’s ranking algorithm includes topic expertise, connectedness, and availability to rank its users for appropriateness of answering questions. Users are able to add tags to any questions they feel are appropriate and may help with classification. The authors highlight the importance of one-to-one conversations over public forums (such as Yahoo Answers!), stating that “*a private 1-to-1 conversation creates an intimacy which encourages both honesty and freedom within the constraints of real-world social norms. (By contrast, answering forums where there is a public audience can both inhibit potential answerers or motivate public performance rather than authentic answering behaviour).*” Users in the system may suggest a friend as a possible answerer. The authors state that as Aardvark is looking at a users extended social network, it provides a key difference to posting questions on one’s Twitter or Facebook in that users outside of a direct social network may receive questions. It could be possible however, that this extended network includes a

large proportion of the entire online community (six degrees of separation²¹). With six degrees of separation, it is thought that any two people on earth are on average only six steps or links away from one another.

The authors further clarify that users are willing to help others with their questions. Users report that it is very gratifying as *‘they have been selected by Aardvark because of their expertise’*, *‘they were able to help someone who had a need in the moment’* and that *‘they are frequently thanked for their help by the asker’*.

Aardvark includes user identities with all messages – this wrapper contains the users *real name, age, gender and location*. Displaying such details may prevent certain questions and answers from taking place, for example, those concerning professional or personal issues or anything of a sensitive nature. However, the authors do present information, opinion and insight into Q&A systems that goes against this idea: *“Overall, a large body of research shows that when you provide a 1-1 communication channel, use real identities rather than pseudonyms, facilitate iterations between existing real-world relationships, and consistently provide examples of how to behave, users in an online community will behave in a manner that is far more authentic and helpful than pseudonymous multicasting environments with no moderators.”* An interesting point, however, it is worth reiterating that this research deals with *real world relationships* and is therefore only appropriate to that form of Q&A – those between two people who know one another directly.

Questions are highly contextualized: where analysis shows that 98.1% of questions are unique and 98.2% of answers are also unique, 45.3% of the question words are content words providing context, which is three to four times as much context in comparison to search engine queries. Details regarding mobile phones are also provided, stating that *mobile users are particularly active*.

Typically, questions are answered within 10 minutes (57.2% of the time). The authors compare this to Yahoo! Answers, where most questions are not answered within 10 minutes. However, this is almost certainly due to the fact that Aardvark has a direct instant messaging interface, while Yahoo! is heavily used with copious amounts of information. Aardvark has a broad range of answerers – 16.9% of users contacted Aardvark to request questions to answer – which suggests some kind of *helpful* or *keen* user classification category could be present in the near future. The authors state that a total

²¹<http://apps.facebook.com/sixdegreesearch/>

of 174,605 distinct tags have been identified to classify questions. This may suggest that users are defining many new tags, and that exact classification of questions is particularly difficult in practice. The possibility of your social network lacking particular expertise and knowledge is also possible.

In summary, Aardvark is a real-time system which helps users gain answers to questions through exploitation of their social network. Aardvark relies heavily on real social connections and identities to motivate and drive question asking and answering. It may be inappropriate to ask certain questions within your social network or with your real identity, and as such, since the official launch of Aardvark, the number one community request is for anonymity: *“allow for users to be anonymous, at times I do not want to have specific responses archived under me”*²².

2.2.3 Real-Time Question and Answer Services

An investigation into the use of markets in Q&A systems by Hsieh and Counts [39] looks specifically at using payment as an incentive to receive answers. The authors present a novel Q&A service that combines payment markets with real-time Q&A. They believe that existing Q&A systems are inefficient and waste the time and attention of the user involved, and require unnecessarily high input from users. Also they do not support the signalling and screening of jokes and non-serious questions, creating potential interpretation issues.

Q&A services act as a valuable alternative to online search engines and help to generate knowledge repositories. The authors again highlight concerns over non-serious and spam questions and stress that *“potential answerers may spend valuable time and attention on these non-serious questions, missing out on more serious questions that really need answers”*. Additionally *“existing Q&A sites are not designed to balance the need of the asker with the availability of the answerer”* which is especially true for real-time systems where *“broadcasting the questions can result in costly interruptions for potential answerers”*. It is suggested that information exchange markets act as a means to solve these problems: *“answerers can then be compensated for their expertise, time and attention”*, where money is used to encourage participation. The authors created two distinct Q&A systems, one with and one without markets.

²²The community section of the site was removed around June 2011, with only a Google cache of the article remaining: whereby the request title was reworded. An original copy of the community request can be seen in Appendix 8.

The authors state that most existing Q&A services are *asynchronous*, using websites as the primary mode of interaction between users (most of which employ some form of reputation-based system). Synchronicity of Q&A systems provides faster answers and faster updates on those answers. Studies suggest that potential problems stemming from the use of real-time Q&A systems are *interruptions* and *inappropriate messages*. The main idea presented is that ordering questions by the amount an asker is willing to pay should allow users to filter out low payment questions which are more likely to be spam.

The authors also highlight that a payment side to Q&A would decrease user intrinsic incentives (altruism) and encourage reciprocity – which is perhaps a disadvantage to the nature of humans helping one another. They define *waste* as the number of answers received before a best answer is found and posit that “*the waste metric provides insight into the efficiency of existing Q&A systems*”. Questions are broken down into several types: *Not a question*, *Factual*, *Advice or Opinion*, and are rated by their seriousness. When users ask questions: “*the question is broadcast to the other users*” which suggests that all users are known and all addresses must be locally known. Additionally, aliases are given to users and displayed on all questions and answers. Scalability becomes an issue here: potentially there could be hundreds or thousands of users you need to source addresses for. The authors discuss the potential for a “*collaborative voting design*”, suggesting that users who are less knowledgeable in a particular subject can still vote on the best answer to a question. They imagine the client application integrating with a website to allow previous questions and answers to be browsed in a knowledge base. As the average level of answer filters (the amount users require to answer a question) is included on screen it would allow an asker to wait for a time when averages are low. This can lead to a form of abuse, where users monitoring levels over time can choose when they should pose questions to lower the cost.

The authors present several hypotheses (H):

H1

The market system will lead to higher average seriousness in questions asked, but will result in fewer total number of questions asked compared to the no-market system.

H2

The market system will incur lower interruption costs for its users.

H3

The no-market system will have faster responses and more answers. In the market system, overall answer quality will be higher, and there will be less waste.

H4

Users of the no-market system will feel a stronger sense of community.

H5

More serious questions will have higher escrow payments.

The idea of aliases for users is presented, but it does not mention anonymous or pseudonymous access. The results show little difference between the systems and still present the usefulness of non-market systems (as hypotheses above). However, this suggests that the no-market system, while admittedly generating more answers, is creating a high number of spam and poorer answers. The authors suggest a significant difference in the level of answer quality: 3.25 for markets and 2.93 for non-market answers out of 5, but in a 1-5 scale perhaps this is not as significant as it first seems due to the limited range available and the fact that these values centre around the mean. One test user commented that they wanted to use the system because it was *“a bit easier and a little more anonymous than the internal mailing list”*. Test user results present encouraging data to suggest that real-time Q&A services are useful if you are notified when an answer is received in real-time. It is stated that *“Questions can then be targeted to a subset of domain experts by asking a high price question within a given topic area”*. They go on to suggest that ‘tokens’ without money or extrinsic goods attached to them could be used for intra-corporate activity or to provide a grade score or level within an organization.

As identified by Hsieh et al. [36] following a study of existing research, it is accepted that people will help one another for free because altruism (the principle or practice of unselfish concern for or devotion to the welfare of others²³), the desire to learn or egoism through the demonstration of abilities, all of which helps to attract users to participate in free Q&A services.

There are clear issues regarding interruptions, interpretation and scalability in existing real-time Q&A services. A scalable solution which provides real-

²³<http://dictionary.reference.com/browse/altruism>

time Q&A is an interesting and novel concept which has many interesting and challenging research questions associated with it.

2.2.4 Expert Retrieval

Expert Retrieval (ER) is an entire research area dedicated to locating experts to help with specific queries.

Previously published ER techniques aim to rank experts for a given query, and in order to do so make use of the availability of [18]:

- A complete list of users to rank.
- Textual evidence in the form of a profile for each user.

Typically the ER community approach to retrieval is to create a global ranking of expertise and route queries directly to known identities. In addition, supporting textual evidence in the form of a profile presents a clear and obvious privacy concern. This concern may increase rapidly with the level of resources used to construct the profile (profiles may make use of all electronic documents and correspondence including e-mails). As such, past ER techniques such as [15, 16, 17, 18, 19, 20, 21] can't be used where privacy is a concern.

This presents an interesting opportunity to explore new means to locate expertise while adhering to the privacy concerns of potential end users.

resource	register	privacy	feedback	Network	Targetted	Medium	Routing
Allexperts	✓	Real Name	expert rating ²⁴	site community	choose expert	web	billboard
AOL Answers	✓	Anonymous	helpful / first to answer counts	site community	categories	web	billboard
answerbag	✓	Pseudonym	positive & best answers	site community	categories	web	billboard
Answers.com	✓	Pseudonym	trust points and report	site community	categories	web	billboard
askpedia	✓	Pseudonym	best answer & report	site community	categories	web	billboard
askville	✓	Pseudonym	–	site community	categories	web	billboard
blurit	✓	–	–	site community	categories	web	billboard
ChaCha	✓	Anonymous	positive & negative	volunteers	categories	web/txt	billboard
Flipster	✓	Pseudonym	none	site community	none	web	billboard
Fluther	✓	Pseudonym	positive & great Answers	site community/socialnet	tags	web	billboard
LinkedIn Answers	✓	Real Name	–	site community	categories	web	billboard
Yahoo! Answers	✓	Pseudonym	positive, negative & best answer	site community	categories	web	billboard
Web Answers	✓	Pseudonym	best answer and report	site community	categories	web	billboard
Aardvark	✓	Real Name	positive feedback & report	site community/socialnet	categories	web/im	interested
Trulia	✓	Pseudonym	positive, best answer & report	site community	categories	web	billboard
Rediff Q&A	✓	Real Name	positive, negative & best answer	site community	categories	web	billboard
oyogi	✓	Pseudonym	positive	site community	topic tags	web	billboard
able2know	✓	Pseudonym	positive, best answer & report	site community	topic tags	web	billboard
Ask Me Help Desk	✓	Pseudonym	positive, best answer & report	site community	categories	web	billboard

Table 2.1: Q&A service comparison.

²⁴Knowledgeability, Clarity, Politeness, Response Time

2.3 Digital Security and Privacy of Communication

Increasingly, data is stored, archived and transported in an electronic format for all manner of purposes. Issues surrounding privacy and security are of constant concern and have been the topic of many recent pieces of research such as [40, 41, 42, 43] which all look at ways to preserve privacy. Work which sets out to improve privacy by being a member of a group rather than a specifically identifiable individual was presented by Wakeman et al. [40]. User location data is obfuscated using techniques presented by Ardagna et al. [41] to protect against user tracking. Both [42, 43] present identification privacy applications via the use of pseudonyms rather than real identities. As so many pieces of electronic data have privacy considerations, and as the volume of personal, organizational and national security data grows and expands, far greater restrictions and concerns need to be addressed.

Encryption has been used to secure transmissions over insecure channels [44], and these founding principles are still used today. Trusted servers need to be adequately protected in order to provide safeguards from threats to user privacy. Recent instances of neglect from the Government with regard to the transportation of electronic data²⁵ demonstrate that this can still occur.

2.3.1 The Internet

Internet Protocol (IP) addresses are used to define a particular host on the Internet. Many of these addresses are dynamic in nature and leased to a particular host at a particular time, for example: home users. As we move towards an IPv6 future, it is possible that each host in the modern world will have a static IP address with an associated host name, and electrical household items will have access to other parts of the local network and indeed remote access to the outside world, providing greater accountability and control. With the ability to clearly identify a specific property, perhaps there will come a time when specific users have a similar address scheme. For example, it is already possible to associate the address of a particular mobile phone with a specific individual.

We are currently running out of the fourth revision of IP: IPv4 [45] addresses, which use 32-bit (4 byte) addresses, allowing up to 2^{32} or 4,294,967,296 possible addresses. Many ranges within the IPV4 address space are reserved

²⁵http://en.wikipedia.org/wiki/20007_UK_child_benefit_data_misplacement

for special purposes and cannot be used for end hosts. To attempt to prevent this problem from accruing in the future and to provide more addresses, IPv6 [46] is in the process of being implemented across the Internet backbone and intermediate routers. IPv6 can support an almost incomprehensible number of unique addresses as it uses 128-bits (16 bytes), allowing up to 2^{128} or 340 undecillion addresses. To put this into context, its been said that “*That’s like a million IPv4 Internets for every single star in the Universe*”²⁶. Not every addresses can be used however as IPv6 has numerous special addresses, such as a loopback localhost address. The Internet still suffers from its share of difficulties and issues, such as those presented in a recent examination of Internet worms (a self-replicating malicious computer program) which focuses on effective spreading techniques [47]. This work reveals a clear insight into the dangers of Internet-based worms, which can spread rapidly, potentially infecting the entire set of nodes at risk within an extremely short period of time. The authors are highly concerned about the possibly disastrous nature of these Internet worms, and recommend a Center for Disease Control (CDC): a form of group or organisation to help combat all aspects of Internet worms. They repeatedly stress the point that worm-based attacks could be used for warfare between nations or to aid terrorism, as has been seen recently in Iran²⁷.

Many papers refer to attackers as enemies. It is clear that the electronic data a country stores is of great importance and the associated knowledge and monetary value of it will be likely targets for military attacks [44].

An attacker who controls millions of nodes on the Internet would possess vast amount of bandwidth, computation and secret or valuable data. The Internet is “*part of a nation’s critical infrastructure*” [47] and an obvious route for attackers. The authors discuss various means to scan for potential victims who are vulnerable to infection from worms. Such techniques could perhaps be used to prevent or alert susceptible computers and their owners to the dangers awaiting them. They also draw attention to the Nimda worm, stating that “*Nimda’s full functionality is still not found: all that is known is how it spreads. But not what it might be capable of doing in addition to spreading...*”. The authors provide details of how worms may efficiently *get off the ground* and infect the most number of nodes as quickly as possible. The following example of two exploits demonstrates rapid infection: pairs of exploits E_s and

²⁶<http://beej.us/guide/bgnet/output/html/multipage/ipstructsdata.html>

²⁷<http://www.homelandsecuritynewswire.com/>

E_c , where E_s is a server exploit and once exploited attaches E_c (the client exploit) to all outputs such as a web page, passing E_s and E_c once again, where the client's browser will attempt to exploit further servers using the E_s vulnerability. It is also suggested that disabling a worm via the same exploit the worm originally abused is possible, and that these kinds of worms should actively disable the vulnerability to protect against this. The authors conclude that the investigated worm "*appears to be able to infect almost all vulnerable servers on the Internet in less than thirty seconds*". With the possibility of exploiting and abusing the current Internet at such a rapid pace, it is worth heeding the warnings of such work and acknowledging that there is still a lot of work to be done to create a safe space.

There is the potential for routing information to be hijacked, meaning that when a particular host is contacted the messages are routed elsewhere – which is extremely dangerous. For example, the rerouting requests for popular web-sites due to hijacked routing information, as happened to `www.youtube.com` in 2008²⁸. Hijackers were able to cause a Pakistan based ISP to announce it was hosting the `www.youtube.com` domain name, meaning that all routing requests for the popular site went directly to the ISP in Pakistan. Luckily the `www.youtube.com` resource does not necessarily require privacy, however, if such a rerouting had occurred with an e-banking resource or similar web service the interception of data-exchanges would be catastrophic.

2.3.2 Anonymous Communication Networks

In some circumstances, sending or receiving data without divulging an identity can be advantageous. With the prospect of ubiquitous computing utilising all manner of emerging data, investigating new ways to push and pull data privately or anonymously seems appropriate.

David Chaum presented "mix networks" [48] (chain of proxies) back in 1981 as a means to provide untraceable electronic mail. He identified possible traffic analysis based attacks, which are the main issue with anonymous communication networks of this type. The research also identifies digital pseudonyms as a means to protect real identifiable individuals. This has been echoed more recently by Evans et al. [42] who use Chaum's work to motivate fresh privacy preserving research. In addition, recent research has been conducted within

²⁸<http://www.bbc.co.uk/blogs/technology/2008/02/>

the Foundations of Software Systems Group here at the University of Sussex [40] who are motivated by techniques to improve privacy for the individual.

Onion routing [49] is an anonymous networking technology for routing data between a sender and receiver through a set path within a fixed overlay network. Original messages are wrapped in several layers of encryption which can only be unlocked or removed by following the chosen path through the overlay network. Each node in the path will remove a layer of encryption to reveal the next encrypted payload and the address of the next node in the route. When the message is finally fully decrypted and forwarded to the intended recipient, the sender of the message is no longer known, all that is present is the identity of the last node that forwarded the message and the payload, removing the link between sender and recipient. In addition to sender anonymity, an example of the message M sent to recipient r through the route z,y,x and PK representing a particular Public Cryptographic Key, the complete onion data structure will resemble: $PK_z(PK_y(PK_x(M,r)),y)$. An example onion routing pathway can be seen in Figure 2.5, hiding the identity of the true source host within a given destination. In this particular example, each host within the onion routing network will decrypt the payload to reveal the next link in the pathway and the next payload (depicted by the black, blue, red and green circle at each host). At the final step in the path the payload is fully decrypted and the end host is contacted directly.

It is worth noting that totally anonymous communications are thought to be impossible [50] as a possible attacker of the system may control any section of the network and even the destination itself. Anonymous routing does however, provide a means to improve privacy and in many practical settings will provide full anonymity.

2.3.3 Sender Anonymity

Typically, anonymous communication networks aim to provide privacy to the initiator of a communication, which is known as sender anonymity. The source may wish to contact a web service or server without directly divulging an identity, perhaps in the form of a network address. A key aim of anonymous communication networks is to provide unlinkability between the sender and the receiver. It should be very difficult to determine if or when a particular pair of hosts made a direct exchange. Figure 2.5 provides an example of onion routing providing sender anonymity.

2.3.4 Receiver Anonymity

Receiver anonymity is a slightly strange idea where the initiator of a communication does not know who they are trying to establish an exchange with. If a network existed where real users were communicating then receiver anonymity could be beneficial, for example during random chat exchanges such as Omegle²⁹, a service which hides the identity of two random chat partners.

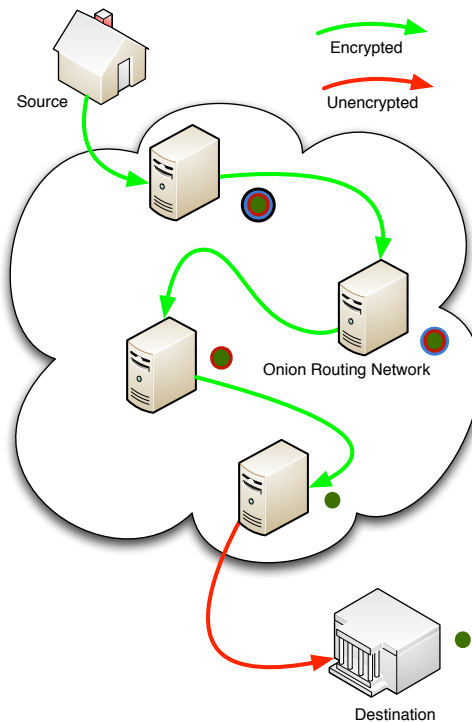


Figure 2.5: Onion routing map.

2.3.5 Location Privacy

Recently, Location Based Services (LBS) [51] have started to emerge and are increasing in popularity. LBS relate directly to a location, for example, an application which provides information on all nearby eateries or hotels when requested by a mobile device. There is growing concern over the potential to track an individual's movements and the ability to uniquely identify user

²⁹<http://www.omegle.com>

query specifics. As a result, the concepts of *cloaking* [52, 53] and *obfuscation* [41] have emerged.

Cloaking hides a specific users query within a set of k additional real users. The notion of k -anonymous queries appears: a query being indistinguishable from the k users within the set. These ideas are feasible; however, they rely on *voluntary participation* from local users in order to form an ad hoc network. They also strongly rely on an abundance of non-beneficial queries and searches in order to establish the initial network. Such techniques may suffer from flaws with regard to device power consumption and will have a strong reliance on accessible local users who are willing to participate in the system. Nevertheless, they are very useful techniques for protecting privacy, especially with the rapidly expanding mobile Internet market.

Obfuscation provides privacy for location data by treating a single location as a range of possible values determined by the accuracy of a given positioning sensor. This range is represented as a circle which can be manipulated to provide location privacy. The actual circular radius can be increased to present a more vague location (in the hope that it will cover some additional number of possible devices), moved (shifting the location's centre), or reduced to produce a less accurate reading (depending on the accuracy of the position sensor). These are very interesting ideas, but they almost seem too trivial for location protection because all of the transformations directly relate to the original location in some form.

2.3.6 Anonymous Opinion Exchange

Research by Kacimi et al. [5] into anonymous opinion exchange over social networks uses hop-by-hop routing to achieve anonymity in the same vein as previous research in this area. Each pair of friends in the social network shares some secret through which they can establish secure connections and communicate privately. This work achieves neither route *selection* nor *direction* – it asks for opinions and not factual information. Authors use k -anonymity techniques to cloak answers by including k fake answers with a query (recognised as a series of 1's), users take a fake answer and replace it with a real answer. Having a fixed-size payload of this type also prevents attacks to gain information from messages of varying size as all queries are the same length as they propagate the network. When all the answers are full or no answer can be given, the query is probabilistically returned to the sender hop-by-hop, making use of a

routing table to remember both forwards and backwards paths. The authors clearly identify that *“the platform is a very powerful attacker since it knows all relations between all users, and it can monitor and store all exchanged messages”*, promoting the possible issue with a *knows all* central authority found within social networks.

Providing anonymous exchanges encourages users to participate even in relation to extremely sensitive or controversial topics. If Q&A can be made anonymous in a similar manner, users may be encouraged to participate more frequently. Within the context of Q&A, users will prefer answers from others who have some expertise or experience in a given subject area. If an infrastructure can be provided to promote the direction and routing of queries between specific users, it follows that the participation of users to the discussion of certain topics (such as health, religion and family & relationships) might increase.

2.3.7 Crowds: Anonymity for Web Transactions

A clear, intuitive solution and an admirably simple technique which offers a degree of anonymity by *blending into a crowd* is presented by Reiter and Rubin [4]. Web servers have access to the Internet addresses of clients and the times and frequencies of user requests; crowds aim to protect user specific identification by hiding within a network collection of individuals. A crowd is formed when a group of users join together to forward and route one another's messages. The general idea of the approach is to probabilistically choose to either make a direct request, or to use a node selected uniformly at random from the crowd to forward the request to. Once a user joins and operates within the crowd it becomes challenging for a web server to directly identify the original initiator of a request as it has been forwarded on by another member of the group.

This crowd system suffers very badly from the predecessor attack, whereby the structure of a web document essentially affords timing attacks. When a user requests a particular website, an abundance of related content may be requested by a typical web browser. If a colluding set of nodes in the crowd receive requests for a particular domain in close succession, the original initiator may be exposed.

The paper states that every crowd member has a shared secret with each other member of the crowd, and this suggests a burden of public and private keys and the associated encryption requirements. Also, the crowds are not fully distributed. There is an element of the system which provides bootstrapping called the ‘blender’. Policies are examined and the authors declare that firewalls cannot be used, and that the set up of new crowd members is not automated and must be done by administrators.

In summary, hiding within a crowd is a valid way to provide some degree of anonymity for network users. Although the specifics of crowds and the nature of typical websites cause problems, passing messages in a proxy-like manner to form a tunnel will hide the true source of a request. This basic principle of passing messages between pairs of nodes is a perfectly plausible technique to disguise question asking and answering.

2.4 Swarm Intelligence

Natural swarm-based techniques have been found to show promising results for ‘dynamic networks’ [54]. Social insects such as ants interact indirectly by modifying their environment and responding to these modifications at a later time. Insects may exhibit ‘collective intelligence’ by simply following rules of direct or indirect interaction to achieve a common goal, such as foraging for food, building a nest or clearing an area of debris.

Stigmergic based approaches exploit the food resources found in environment by interacting indirectly without global knowledge. These principles can be used to support resource locating in computer networks by increasing the probability of routing to better areas of the network based on past interactions and performance.

2.4.1 Stigmergy

Stigmergy is derived from the Greek words ‘stigma’, to sting (mark or sign), and ‘ergon’ meaning work or action, and is used by ants while foraging for food. Ants of various species exhibit trail-laying and following behaviours when foraging for food. They deposit chemicals (pheromones) as they travel to and from the nest in various directions. These pheromone trails are then followed by other ants also on the quest for food. The first ants to return to the nest

having successfully found a food source, will doubly reinforce pheromone trails taken to said source, thereby encouraging new foraging recruits or unsuccessful ants towards the source.

Artificial ants in computer programs which follow the same basic principles and behaviours have been used to solve an array of combinatorial optimisation problems [54]. Often ant-based solutions are as good as general heuristic solutions and in some cases, when combined with heuristics, can achieve remarkable results. Ant-based algorithms are known to be particularly good at dealing with problems that change over time.

2.4.2 Foraging Strategies in Ants

In 1989 it was proved that path selection towards a food source is definitely based on self-organisation for an Argentine ant (*Linepithema humile*³⁰) species [55]. A simple experiment was conducted involving a nest with two bridges of equal length (A and B) to a food source (see Figure 2.6). Ants chose a path towards the food source via one of the bridges. It emerged that one path will be selected more frequently due to the stronger pheromone trail. A model of this process was developed, where the probability of selecting a particular path is directly proportional to the number of ants that have used it. If A_i and B_i represent the number of ants that have used branch A and B, then the probability of ant $i+1$ selecting branch A(B) can be seen in Equation 2.1. This equation shows that greater levels of pheromones on branch A increases the probability of branch A being selected. In this experiment, the values of n and k were fitted empirically with the values $n = 2$ and $k = 20$. Parameter n is used for the degree of non-linearity (if large, the slightest difference in pheromone levels will result in higher probabilities of selection). The parameter k is used to specify the degree of attraction; the greater k is the greater the levels of pheromones that are required to make the selection non-random in the form of a bias. It is known that ants choose a path based on a function of the cumulative number of ants who have taken that path, which may be dynamic and change over time.

$$P_A = \frac{(k + A_i)^n}{(k + A_i)^n + (k + B_i)^n} = 1 - P_B \quad (2.1)$$

³⁰Formerly *Iridomyrmex humilis*

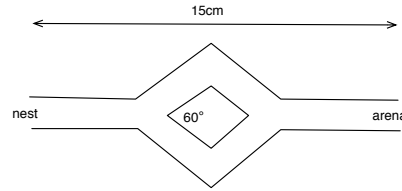


Figure 2.6: Binary bridge experiment.

The equal length bridge experiment was expanded such that one branch was longer than the other by Goss et al. [56] (See Figure 2.7). It was shown that the shortest branch is selected most often as the volume of ants taking the same shortest length path to and from the food source will reinforce the pheromone trail on that path due to more journeys being taken and in addition, more ants will be recruited to forage for food. Unfortunately, in this work it appears that ants have the potential to begin reinforcing the longer path if it is probabilistically chosen by the colony initially, hiding the shorter route. In addition, if the shorter path is presented at a later stage in the experiment, the longer path may already have such a strong trail that the new path is never signposted as the shortest path. It is worth noting that the selection process used by the ants is not based on individual ants recognising the differing lengths between branches in the experiment, but rather that a collective self-organisation process takes place indirectly between ants via the pheromone trails produced during the foraging process. Deneubourg uses the analogy that pheromone trails in ants is similar to that of any trail formation, for example, as seen by animals in grasslands or even by students on a snowy campus. The greater the number of animals which have used a particular path, the more disturbance there is to the environment, and the more trampled grass there is the more likely it is that additional animals will choose the same trail. The authors recognise that due to the use of sand as the experiment base, tracks or ‘valleys’ do emerge from the multitude of ants using the same path, and perhaps this has some influence on the route choice. However, it is clear that without the pheromone trail the ants will not choose a particular route. Interestingly, experiments have shown that pheromones are not well understood chemically, all that is known is that they last for several hours but can persist for many months.

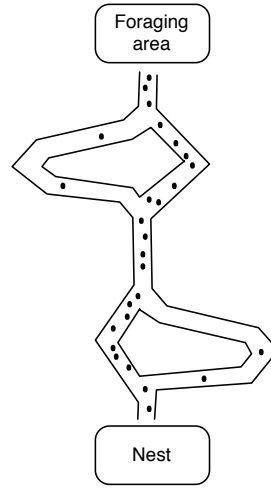


Figure 2.7: Short branch experiment.

An emergent factor of the ant foraging process is that the shortest path between the nest and the food source is likely to be located and maintained. A path optimisation problem such as the Travelling Salesman Problem (TSP) is relevant here. TSP aims to find the shortest tour connecting n cities, where each city is visited exactly once. There are two main forms of the problem where the distance between any two cities is either symmetric or Asymmetric Travelling Salesman Problem (ATSP), however, the solution is achieved in the same manner regardless of the form. In ATSP, the path cost between two cities depends on the starting location. Research found that a stigmergic technique could be used to solve the TSP problem [57] with results comparable with existing general purpose heuristics. However, when the technique was combined with a local search in 1997 it performed exceptionally well [58].

Although the ants are unable to optimise the chosen path naturally every time, they can find the food source and indicate its location to the colony. This is an important observation as these techniques can be used to locate resources. In nature, ants are likely to benefit from focusing the combined efforts of the colony on a single location at any one time and this relates to the techniques they utilise.

2.4.3 Key Elements of the Stigmergic Technique

There is a general template [54] for applying the stigmergic technique to a range of problems, using the following categories:

Problem Representation

This allows ants to build and modify pheromone trails on a graph, which includes information of past path performance.

Heuristic Desirability

This relates to the set of possible edge choices available at each branch. It indicates if a specific branch is more desirable for a specific solution and is often included in the probabilistic transition rule.

Constraint Satisfaction

This allows for the construction of possible solutions. It defines when a solution has been reached and represents the end of a specific run of the algorithm.

Pheromone Update Rule

This specifies how to update pheromone strengths on the edges of the problem graph. The Pheromone Update Rule is used to modify the environment to support better solutions by promoting the routes which have presented the best solutions to date.

Probabilistic Transition Rule

This functions on the pheromone trail and the heuristic desirability of an edge. It is used to determine which path is chosen when a choice is made available.

2.4.4 Security and Privacy

It is well known that stigmergy is able to provide a degree of privacy for the source and destination of a communication [59]. It has the potential to require no detailed information about plans for the delivery destination or the true source of a message. With routes being made entirely from pheromone strength and path selection based on a random mechanism, the detailed specifics of message delivery and routing are unpredictable to the end user. A recent paper presented a comparison of numerous stigmergic routing protocols, all

of which included complete path identity information [60]. A lot of existing research looked at during this research work into stigmergic routing places little emphasis on privacy concerns. This is surprising because it has the potential to protect privacy to a degree. Tables 2.2 classify ant routing approaches according to the way in which they record identities of observed nodes.

Additional techniques, such as TOR³¹, can be used to guarantee privacy by hiding the true source address of a node behind an extra anonymous routing layer.

2.4.5 Telecommunication Stigmergy

A well known paper which uses the principles of ant colonies within the British Telecom Synchronous Digital Hierarchy (SDH) telecommunication network proved successful and popular with the academic community [7] (see Figure 2.8). Schoonderwoerd et al. [7] investigated the use of stigmergy on a telecommunication network to attempt to maximise the performance (rate of accepted calls) at runtime. Agents called ants are used to explore the network and deposit pheromones on the routing table entries of nodes as a function of the congestion at nodes. Routing policies are based on the pheromone trails in the network and these adapt during the lifetime of the network. Systems like this are known as ant-based control (ABC) systems.

The network is represented as a graph $G = (N, e)$, with N representing the nodes and e representing bidirectional links within the network. Each $node_i$ has routes to k (one or more other nodes) in the network, and is also associated with a total capacity C_i , spare capacity S_i and a routing table which includes an entry for each available neighbour (1.. k). Pheromones are represented as routing probabilities. Each routing table contains a pheromone table for every possible destination in the network, and each table contains an entry for each local neighbour. **The authors state that ‘an N-node network uses N different kinds of pheromones’ – this is an interesting view and provides a means for having multiple trails for different purposes to each neighbour.**

Two issues are highlighted: *the blocking problem* (where a previously found route is no longer available, which can take a long time to resolve) and the *shortcut problem* (where new and shorter routes suddenly appear, but new

³¹<https://www.torproject.org/>

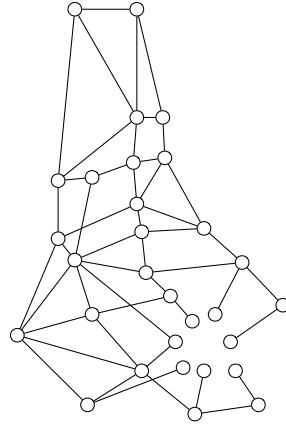


Figure 2.8: British Telecom synchronous digital hierarchy (SDH).

routes will be hard to discover as old routes have been strongly established with pheromones such that they are almost always chosen). The authors aim to overcome both of these problems.

Ants are generated randomly at nodes in the network, also picking a destination node at random. Each ant passing through a node updates the pheromone entry to the source node laying the kind of pheromone associated with the node they were launched from. Probabilities and pheromones can be assumed equivalent in this work; pheromones are updated by incrementing values by a change in probability while normalising to make sure the entire collection of values sums to 1.

Changes in probability used to update pheromone tables are a function of the ant's age (how many steps it has traversed), influencing the system to lean towards paths of shorter length with younger ants on them. Ants which encounter congestion in the network are delayed, causing them to age. The authors state that noise or randomness is required to create *exploration* of the network: *"every time step an ant has probability F of choosing a purely random path, and probability $(1-F)$ of choosing its path according to the pheromone tables on the nodes"*. This idea is reinforced by the work of Deneubourg in 1990: *"Rather than simply tolerating a certain degree of error, it can be desirable to deliberately add error where none or little exists."*

Finally, the authors note that this technique is likely to require more computation than traditional methods due to the extensive use

of random generators. The space requirements are also increased in comparison to traditional routing tables, as nodes need more space for allocating pheromone tables. Today however, with the ever-increasing computation and storage abilities of modern devices, these specific problems are becoming less relevant.

2.4.6 Stigmergy in Computer Networks

Stigmergy has also been used in the area of Computer Science and routing in computer networks. Essentially, these are very similar to telecommunication networks, and so the transition is a logical one.

Work surrounding the idea to use ant techniques and principles for forwarding queries in P2P networks has appeared in recent years. One such study by Michlmayr et al. [61] plans to use multiple types of pheromones associated with various categories from an ontology to build pheromone strength values. Queries will attempt to follow those paths with the strongest pheromone values in the associated query category. The authors aim to “...*maximize the number and the quality of query results while minimising the overhead necessary for management of routing tables*”. Authors identify that ant-based techniques are suitable for P2P networks for two key reasons:

Decentralisation

As communication between ants is indirect and takes place solely through the modification of pheromone trails in routing tables, ant-based methods do not require any global knowledge which is directly applicable to decentralised systems such as P2P networks.

Dynamic Behaviour

Pheromone trails can update dynamically to reflect paths as nodes join and leave the network, typically seen in all P2P networks as a key feature known as churn.

Multiple pheromones are used for each routing table entry to represent multiple concepts of queries. While the queries in the P2P networks themselves represent ants in the network, *backwards ants* are used to provide feedback by updating pheromone values in accordance with the current solution. These special ants are generated once the destination node has been reached. Popular

queries require optimised paths and therefore the use of ants in this approach is highly applicable.

A novel approach to exploring and exploiting knowledge of the network as seen in this work is to take an average across several pheromones when a query is related to more than one concept. The average pheromone values are then used for path selection rather than single concepts.

The key difference between P2P systems and Q&A networks is that P2P systems require global knowledge to locate data appropriately. In addition, P2P systems are typically serving data files from the network rather than real human responses where human attention and time are consumed.

Other work has investigated the use of stigmergy within mobile wireless ad hoc networks [10]. The authors aim to use biologically-inspired techniques to maximise packet throughput while directing traffic towards regions of high throughput and aiming to promote routing robustness and scalability. The authors confirm that self-organisation techniques are based on the following key principles:

1. Positive Feedback
2. Negative Feedback
3. Randomness
4. Multiple Iterations

This set of principles neatly describes the requirements of a stigmergic approach at a high level. The termite technique is based on and adapted from the well known ABC approach [7], which uses positive feedback in the form of pheromones, and negative feedback in the form of exponential pheromone decay. The use of pheromone *ceilings* and *floors* are used to limit the minimum and maximum values of routing table pheromones and to prevent extreme differences in entries from spoiling the routing process. All packets in this system include where the packet was just sent from, but also the original source address and information detailing how quickly it has travelled through the network. The simulations of the approach described appear to be for a small network of fifty nodes and run for 300 seconds.

In summary, stigmergy can be used in communication networks to aid the routing of data around real world networks. As it is able to provide emergent

routes without specifically requiring node identities, and is aware of churn issues, it seems a most promising technique for routing data in decentralised networks where routes can emerge over time.

reference	feedback	churn	resource	pheromones	privacy	application
Schoonderwoerd et al. [7] ((year?))	[+]	—	call failure	per node	none	ROUTING
Subramanian et al. [8] (1997)	[+]	failures	delivery time	per node	—	ROUTING
Barán and Sosa [9] (2001)	[+]	failure	delivery time	per node	—	ROUTING
Roth and Wicker [10] (2003)	[+]	mobility model	throughput	per node	—	MANET
Baras and Mehta [12] (2003)	[+]	mobility model	goodput	per node	—	MANET
Heissenbüttel et al. [11] (2003)	[+]	—	—	per node	—	MANET
Michlmayr [62] (2006)	[+]	—	results	per keyword	—	ROUTING
Berlioli et al. [63] (2004)	[+]	—	—	per node	—	SATELLITES
Yun and Zincir-Heywood [13] (2004)	[+]	failures	throughput	per node	—	ROUTING
Caro et al. [14] (2005)	[+]	mobility model	delivery	per node	—	MANET
Hoh et al. [64] (2006)	[+]	mobility model	delivery	per anycast	—	ANYCAST
García and Pedraza [65] (2008)	[+]	mobility model	delivery	per node	—	MANET
De Rango and Tropea [66] (2009)	[+]	mobility model	energy	per node	—	MANET

reference	route selection	update	bounds	nodes	type
Schoonderwoerd et al. [7] ((year?))	$P^i(j) = \frac{p(j) + \alpha l_j}{1 + \alpha(N_k) - 1}$	proportional	[0.0, 1.0]	—	proactive
Subramanian et al. [8] (1997)	$P_{i,d} = \frac{(P_{i,d})}{\sum_{j=1}^n (P_{j,d})}$	proportional	—	31	proactive
Barán and Sosa [9] (2001)	$P_{i,d} = \frac{(P_{i,d})}{\sum_{j=1}^n (P_{j,d})}$	proportional	—	55	proactive
Roth and Wicker [10] (2003)	$P_{i,d} = \frac{(P_{i,d} + K)^F}{\sum_{j=1}^n (P_{j,d} + K)^F}$	linearly	[0.1, 1000]	50	proactive
Baras and Mehta [12] (2003)	<i>deterministically</i> (max)	proportional	—	20	proactive
Heissenbüttel et al. [11] (2003)	$P_{i,d} = \frac{(P_{i,d})}{\sum_{j=1}^n (P_{j,d})}$	proportional	—	—	proactive
Michlmayr [62] (2006)	deterministically (max)	proportional	—	1024	reactive
Berlioli et al. [63] (2004)	deterministically (max)	proportional	—	—	reactive
Yun and Zincir-Heywood [13] (2004)	$P^i(j) = \frac{p(j) + \alpha l_j}{1 + \alpha(N_k) - 1}$	proportional	—	55	proactive
Caro et al. [14] (2005)	$P_{nd} = \frac{(T_{nd}^i)^{\beta}}{\sum_{j \in N_d^i} (T_{j,d}^i)^{\beta}}, \beta \geq 1$	proportional	—	1500	proactive
Hoh et al. [64] (2006)	$P_{nd} = \frac{(T_{nd}^i)^{\beta}}{\sum_{j \in N_d^i} (T_{j,d}^i)^{\beta}}, \beta \geq 1$	proportional	[c*5, c*8]	150	reactive
García and Pedraza [65] (2008)	deterministically (max)	proportional	—	50	proactive
De Rango and Tropea [66] (2009)	deterministically (max)	proportional	—	50	proactive

Table 2.2: Stigmergic comparison.

2.4.7 Stigmergic Technique Conclusions

A networked collection of computers can use virtual pheromones within routing tables to route data based on probability and predefined rules. As the network witnesses interactions and learns of better performing routes the routing tables can be updated to reflect this knowledge.

Each node in the network may have its own local collection of pheromones and rules which can be used to update routing table entries accordingly. A simple set of rules can be used to update pheromone values, perform evaporation and probabilistic route selection. In addition, there is no need to include the final destination or source address of a question and answer exchange.

2.5 Peer-to-Peer

Peer-to-Peer (P2P) is a distributed computer architecture which facilitates the direct exchange of information and services between individual users (peers), rather than relying on a centralised server. It forms the basis of most traditional distributed computer systems, permitting each peer node to act as both a client and a server and consuming services from other available peers, while simultaneously providing its own piece of the entire service to the rest of the network. Peers within a P2P network engage in direct exchanges with their P2P neighbours in order to submit requests and serve responses.

It is known that P2P has many advantages including: scalability, high resource availability, low cost, no central authority and robustness. The consequence of using this architecture is that the quality and usefulness of the services on offer rely entirely on the members of the group that exist within it.

The definition of what exactly constitutes a P2P system is broad and makes exact classification troublesome. For example, despite P2P systems being applauded for having no centralised authority, in reality, many existing P2P applications rely heavily on such systems. The following definition by Risson and Moors [67] is well suited to classifying P2P systems:

Peer-to-Peer systems are distributed systems consisting of interconnected nodes able to self-organise into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable con-

nectivity and performance, without requiring the intermediation or support of a global centralised server or authority.

The power of P2P is made clearly apparent when considering Metcalfe's Law³², which stresses the significance of the number of potential connections found within communication networks. The number of unique connections found is n^2 in relation to the number of network nodes n , the number evaluating to $n(n-1)/2$. Therefore, many different routes between any pair of nodes can be taken advantage of. Network entities may or may not end up communicating directly, however, routes do exist between any pair of nodes connected to a P2P network through emergent pathways, which may be either relatively cheap and short, or long and complex.

With the enormity of the modern global Internet and the prospect of expanding it even further with the introduction of IPv6 [46], this will allow for a very large number of devices to be specifically addressed. As such, P2P will continue to offer the potential for highly scalable and efficient networking technologies for years to come on a multitude of devices.

2.5.1 Peer-to-Peer Networking

P2P networks form overlays on top of TCP/IP to support networked decentralised services. A brief overview of the key features of P2P are included below.

2.5.1.1 Neighbours and Connectivity

A peer's neighbours form direct communication links by providing possible routes for forwarding requests and messages, and provide connectivity to the entire P2P network by acting as gateways to collect and receive responses. Peers need to keep track of their neighbours to ensure that they remain connected to the P2P network. As such, peers may need to drop or request new neighbours throughout the duration of their P2P session to ensure proactive connectivity.

³²http://en.wikipedia.org/wiki/Metcalfe's_law/

2.5.1.2 Network Churn

The action of peers arriving and leaving a network is known as network churn. It is important to recognise that peers will arrive at a certain point, and begin to participate for some period of time (session duration), possibly making requests and serving other users, before eventually leaving, although they may return at a later time. High levels of churn indicate many user arrivals and departures, while low levels indicate longer peer session durations with less frequent arrivals and departures.

By nature, the level of churn will vary between different P2P applications. For example, the popular BitTorrent³³ network may consist of a handful of peers who are available for long periods of time to serve individual pieces of files, while many peers will appear and leave shortly after file downloads have completed [68].

P2P networks may perform differently with different levels of churn, so it is important for designers to consider expected levels in order to provide an appropriate service. A simple example would be the replication of critical files within a chaotic, high churn environment, to provide greater data availability for the network.

2.5.1.3 Bootstrapping

Peers need some means of gaining access to and becoming a participant of a given P2P network. This process is known as bootstrapping. It is likely that peers will bootstrap using some kind of centralised resource, returning an entrance point into the network in the form of a set of active network member addresses or other centralised repositories.

A bootstrap protocol already exists to locate a server when the bootstrap address and even the local addresses are unknown [69]. Such a protocol allows data to be requested from an unknown bootstrap server by broadcasting over the local network address 255.255.255.255. Although providing the appropriate functionality, this is not appropriate for deniable focused systems due to the announcement of actively wishing to participate.

³³<http://www.bittorrent.com/>

2.5.1.4 Routing Tables

Each node in a P2P network will keep some information about some subset of the entire network: this small proportion of the network is referred to as the node's *neighbours*. Nodes will generally keep a reference to these names in the form of an address, and will often also store meta-data to help improve routing options. A routing table may associate a set of end-host destination addresses with each neighbour to aid route selection to a given destination.

2.5.2 Example Peer-to-Peer Applications

There are a multitude of P2P systems and applications which have been used in the past or are in use today, with many receiving mainstream attention. Below follows a short introduction to some popular systems.

Napster

A distributed P2P file-sharing system from the 1990's Napster provides the functionalities for MP3 music files to be located and downloaded from other online members. The system makes use of a central system to manage user search requests, with all traffic flowing through the central site except downloads which are direct. Napster presents a solution to searching a large collection of hosts without the bottleneck of serving file download requests. It was very successful but it is brokered by a know-all central authority and control who is able to record all user activities and control many aspects of the service.

Gnutella

This is a P2P file-sharing utility that allows files (such as music, images and documents) from other members to be searched and downloaded via distributed search. Gnutella uses a flooding approach to propagate search queries across the network. There are many different strands of Gnutella including but not limited to: Morpheus, BearShare and LimeWire. Many of these Gnutella-based applications have been used for illegal file sharing.

The traditional Gnutella protocol consists of the messages: Ping, Pong, Push, Query and QueryHit³⁴, and can be broken down into the following functionalities discussed below.

³⁴http://www.sans.org/reading_room/whitepapers/threats/overview-gnutella_455

Bootstrapping: a ‘Ping’ (identified by a unique Globally Unique Identifier (GUID)) can be used to announce presence in the Gnutella network. Any node who receives a ‘ping’ message responds with a ‘Pong’ and forwards the ‘Ping’ to its own neighbours who will also reply with a ‘Pong’. A ‘Pong’ contains information about the host including an IP address and port number and some meta-data associated with the files being shared. A ‘Pong’ does not have to return directly to the source of a ‘Ping’ and may follow the original multi-hop route taken. Having sent a ‘Ping’ into the network, a host will quickly become aware of its active peers.

Search: hosts may send a ‘Query’ into the network via all known links (‘Pong’ responses). A ‘Query’ will contain some search terms and will be used to pattern match it against local files. A host who receives a ‘Query’ will first attempt to match against the files being shared locally, and then forwards the ‘Query’ to its own neighbours and so on. Time-to-Live (TTL) values are used to stop messages lingering indefinitely in the network and GUIDs are used to prevent ‘Query’ loop and cycles emerging.

Download: any host which has a local file matching a ‘Query’ will reply with a ‘QueryHit’ via the original multi-hop route containing the hosts address and information about the files available. Files are then downloaded directly via HTTP GET requests if possible, using the ‘Push’ message as a final resort to download directly from the end host via the intermediate route.

Freenet [6]

This system allows for anonymous distributed storage, search and retrieval of files. Data is made available while it is actively being used and popular, and old unused content is eventually replaced. Freenet does not use a centralised control or administration. The encryption and relay of messages makes it difficult to determine who is storing data, where it is being requested from and where it will be stored.

Freenet nodes allocate disk space to store data from the network to create a cache. When users store data on the network it is broken down into blocks and encrypted, replicated and stored on various nodes. The

network does not need entire files to be available at all times from a single node, as long as all blocks are available from some location.

Freenet provides plausible deniability to data publishers and consumers without any need for preferential routing mechanisms or file quality.

2.5.3 Advantages and Disadvantages

P2P systems are extremely powerful as they provide network resources (such as CPU-cycles, memory and storage facilities) which increase with the number of active hosts. As P2P services are distributed across the network, there are little or no setup costs, reducing overall costs for development and service maintenance. They also encourage participation as network nodes operate as both clients and servers.

P2P systems do not suffer from a single point of failure like centralised systems as all core functions are distributed across the network. In addition, there is no single performance bottleneck to slow down or restrict the operation. Similarly, there is no central authority to dictate the rules of the service, providing freedom to the end users. However, there is also no central body to control unwanted behaviours such as authenticity, viruses and copyright infringement.

2.5.4 Routing in P2P Networks

The work presented by Tempich et al. [70] is highly relevant to the themes and ideas found in this thesis. The key idea behind *REMINDIN'* is that peers observe which queries are successfully answered by other peers and in turn use this information to select peers to forward their requests to in the future. The only difference is that it routes queries to entities within the network which already have results to a query, for example a database, documents or e-mails.

Insurmountable obstacles to routing in the P2P environment are highlighted, including the openness of the domain, the fact that peers do not inherently know where to find relevant information and that deciding which information to make use of can be difficult.

The routing in this work is based on a social metaphor. It mimics what a person is doing in a social network to get answers to questions, namely retaining meta-information about what other peers in the network know about,

perhaps not directly, but from observed communications. It also asks one or a few of the peers how they estimate their own coverage and the reliability of information about particular topics.

Peers have confidence values representing the “*confidence assigned to a remote peer concerning a particular statement*”. It is also anticipated that some peers may be more knowledgeable than others and that the more knowledgeable peers are those that in general provide a lot of information.

The authors make the following useful assumptions based on the social metaphor: 1) “*A question is asked to the person who one assumes that he best answers the question.*” where best in this context represents the most knowledgeable peer. 2) “*One perceives a person as knowledgeable in a certain domain if he/she knew answers to our previous questions.*” 3) “*A general assumption is that if a person is well informed about a specific domain, he/she will probably be well informed about similar topics e.g. the next more general, topic, too.*” 4) “*To quite some extent, people are more or less knowledgeable independently of the domain.*” 5) “*The profoundness of knowledge that one perceives in other persons is not measured on an absolute scale*”. This routing approach uses the above observations of the ways that people act in order to solve queries and to investigate and improve the performance across a number of key metrics.

Queries in the network are sent to a *subsection* of selected peers based on the social-metaphor observations, and the *eventual answers are returned directly to the querying peer*. A relaxation algorithm (weakening the peer selection requirements) is used to aid peer selection when the choice of peer is not immediately obvious. The above social metaphor assumptions are used to control query routing. The complete route taken by a query is stored along with it and any peer may become visible to other peers when its identifier is known. Interestingly, a dynamic network setting in which nodes join and leave the network (churn) is not included in this work and as its an important aspect of P2P networks it should have been considered.

Evaluation metrics in this approach are 1) Recall (R); the proportion between all relevant statements in the peer network and the retrieved ones: $R = \frac{|retrieved|}{|relevant|}$. 2) Network Load, in the form of messages per query. It traces to what extend the network is being flooded by one query. 3) The Average Number of Hops, which can indicate how goal-orientated (how quickly a destination is located) the query routing is and how fast an answer may be

returned. The authors use these three metrics to assess the efficiency of the approach and results indicate that efficiency is indeed improved against more naïve approaches in terms of recall, network load and number of hops.

To recap, social metaphors can be used to aid routing in networks, using observed performance of routes and iterations in relation to a clear ontology. It is interesting to find new decentralised means to route queries to achieve the best performance possible with little or no existing knowledge of the peers. This work investigates routing towards peers who already have the answers and results to queries. Considering real-time responses from humans would be far more interesting with real networking behaviours such as network churn.

2.5.5 Peer-to-Peer Simulation

The computer devices sitting at the edge of the Internet have the potential to provide vast resources and computation power, all of which will be readily available for the exploitation of P2P applications. However, P2P networks may consist of thousands of unique entities with the potential to expand above and beyond in terms of active network participants. The evaluation of such huge systems in active operation with real users and software is a daunting and complex task. This has been demonstrated in several existing works all of which employ P2P simulation [71, 72].

There exist *many* different P2P simulation technologies, many of which aim to provide a P2P infrastructure with an abstracted networking model on which users can build. As shown by Naicken et al. [73], much research is evaluated via the use of bespoke one-time P2P simulator technologies. This is due to the difficulty of learning to use and understand existing simulators. The various P2P simulators utilised in academic work make it incredibly difficult to verify whether the results are legitimate.

In its most simplistic form and ignoring all underlying networking implementations and protocols, a P2P node consists of an incoming and an outgoing ordered message queue which is processed and manipulated over time. A node is aware of a number of other nodes within the network, known as *neighbours*. Nodes pass messages to and from their neighbours, manipulating message queues. For example, if node X wishes to pass a message M to node Z, then X aims to find some route to Z such that $X \rightarrow^* Z$ via some route of

intermediate nodes. The simulator should allow for realistic behaviours to take place such as network churn and bootstrapping.

2.6 Summary and Research Direction

Q&A is a popular service which allows users to appeal to a broad range of answerers without requiring specific expertise, time or effort. In most cases, these services provide answers by presenting questions to the general public or associated digital community with little regard for the amount of time users may spend examining and potentially answering questions (see Table 2.1). A question may receive large amounts of attention from users but not be answered adequately. Existing research has investigated the reasons why questions do not receive answers in Q&A services, suggesting that it may be associated with users being afraid of expressing themselves. With new Q&A services utilising social networks and the increasing presence of real identities, a range of topics may not be discussed, thus reducing the potential utility of the overall service.

Stigmergic-routing tactics can be used to direct questions towards knowledgeable members of the network. The stigmergic algorithms map neatly onto routing within ad hoc networks and the protocols which run on them. Q&A services work well for solving information needs, but rarely take into account user privacy. Stigmergic routing is a suitable technique in decentralised networks, however, it does not typically take into account the privacy of source and destination node identities, (see Tables 2.2) even when it is particularly suited to this task. Several routing strategies exist for anonymous systems, yet they fail to make ad hoc choices based on their current active user base and do not pick up the interests and expertise of users. An anonymous routing solution for Q&A appears to have been overlooked in past literature.

The design and evaluation of a deniable Q&A network which works over decentralised ad hoc networks is a novel system which warrants further investigation and research. For the reasons highlighted above, this thesis will investigate a more targeted approach. The goal will be distributing the Q&A service across ad hoc networks and using routing to locate preferred answerers. The routing will attempt to reduce the consumed attention of those users who cannot help with the answering process. Finally, to attempt to counter user's concerns the distributed Q&A service will strive to be plausibly deniable, hiding user identities within the networked crowd of individuals.

User Modelling

3.1 Background

With the rise of Web 2.0 we can make use of publicly available digital footprints from web-based services to model the activity of users. Modelling Q&A users allows us to simulate some of their traits and behaviours, which is helpful in the evaluation of approaches for routing in Q&A networks. It is infeasible to model the exact properties and behaviours of individual users, but we may observe a population of users as a whole and aim to model properties of the entire population.

In this chapter we perform an analysis of Yahoo! Answers users and their Q&A, as well as an analysis of data collected from `twitter.com` during the course of this study. The aim of this chapter is to identify the appropriate user traits and their distributions in order to aid Q&A user modelling, with the intention of using the sampled users to simulate realistic Q&A networks.

3.2 Yahoo! Answers dataset

Yahoo! Answers provides a means for people to appeal for help with any question (as discussed in Chapter 2).

A Yahoo! Answers dataset [74] was acquired from Yahoo! via a formal request and data sharing agreement (Appendix 8) for non-commercial research purposes. The dataset consists of the entire Yahoo! Answers corpus from its creation in 2005 up to the 25th October 2007. The corpus boasts an impressive collection of 4,483,032 questions and their associated answers. Prior to requesting the dataset, a PHP script which uses the Client for Uniform Resource Locators (cURL) and regular expressions to extract category names,

URLs and IDs (from the public Yahoo! Answers HTML source code) was created to attempt to parse the Yahoo! Answers website. An interesting discovery was made while extracting data from Yahoo! Answers without using the Application Programming Interface (API) – if Yahoo! detects that the site is being parsed automatically or systematically in a manner not possible by a human being, the custom HTML error code 999 is returned. This took some time to detect using cURL and, even when using techniques to disguise the cURL calls via explicit user agent headers, many of the requests returned an error and needed to be remade. Due to the custom Yahoo! error codes, a repetitive recursive process must be used to brute-force the collection of categories, which unfortunately is a process that takes a considerable amount of time. Therefore, it would be both difficult and extremely time consuming to parse the website directly in order to gain the same volume of information found in the dataset. This level of prevention to restrict crawling the website is due to past attempts to collect data from Yahoo! Answers and aims to both reduce website traffic and limit the direct extraction of data for commercial use.

The dataset consists of several supporting files and the main data is compressed as a gzip¹ file (`Full10ct2007.xml.gz`) which is 3.71GB in size. Once uncompressed, the dataset is stored in a single XML file called `Full10ct2007.xml` and increases to 12.26GB in size.

Clearly this is a large dataset which in its raw XML format would be difficult, if not impractical, to process and use directly. It was decided that a Structured Query Language (SQL) database, when correctly optimised, could handle this level of data. The XML corpus was extracted using a bespoke Java program utilising the Streaming API for XML (StAX). A StAX² approach was adopted to avoid having to load the entire Document Object Model (DOM) into memory, which due to its size, would have proved unachievable using the computing resources available at the time of this study. The XML structure can be seen in Appendix 8.

3.2.1 Database Structure

Below is a diagram showing the representation of the dataset within the database (Figure 3.1) which has been specifically designed to organise the

¹<http://www.gzip.org/>

²<http://download.oracle.com/javase/6/docs/api/javax/xml/stream/XMLStreamReader.html>

corpus in a more useable way.

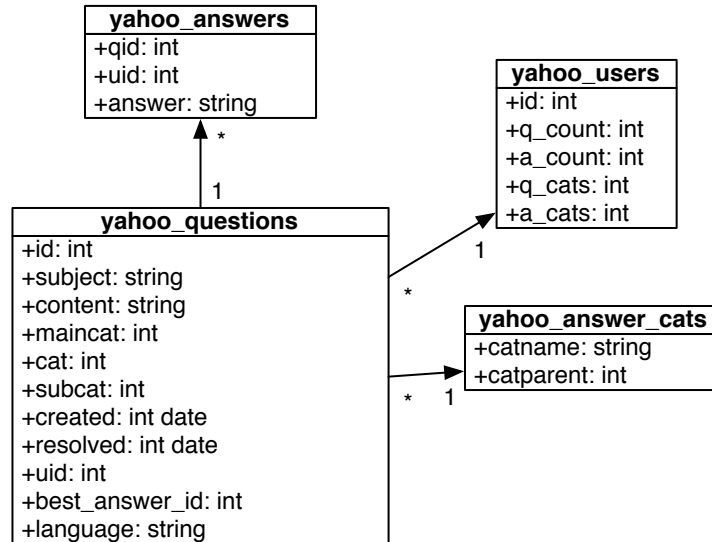


Figure 3.1: SQL database structure of Yahoo! Answers corpus.

In essence, the SQL tables package together the data from the corpus while organising and restructuring several features to increase utility, for example: creating a new category table which uses a numeric rather than a string reference. Another example is the use of tables and fields following data processing, to allow the number of questions and best answers each user has supplied to be determined, and for these findings to be included in the users table (*yahoo_users*).

Due to the size of the database, queries take a significant amount of time to process and return. To improve the situation for analysis, indexes³ were used to add additional structures to the database and improve common search options. The *yahoo_questions* database has indexes for many of its fields, including question identification number, authoring user, best answer identification number, language and category.

³<http://dev.mysql.com/doc/refman/5.0/en/mysql-indexes.html>

3.2.2 Languages

Within the dataset, language was combined with location to allow for evaluation of the data in comparable units. In total, there were 85 possibilities, of which, 74 held less than 1% each.

There are over 30 possible languages which exist in the dataset, out of which, only 6 languages each have over 1% of the questions. Just 6 possibilities make up over 99% of the dataset. The majority of the dataset is defined as being “en-us” (English from the United States) and contributes 90% of the dataset.

From this point onwards we will only be considering questions that were asked in the English language as it represents the bulk of the dataset (90%). Investigating the English questions and answers also allows for a direct understanding of the words, category and supporting information used without the need for a translator.

3.2.3 Date Ranges

The dataset includes the entire corpus of data from two years, 2005 and 2006. The majority of the data is from the year 2006 (See Table 3.1).

Year	Number of Questions
2005	75,941
2006	3,472,328

Table 3.1: Questions asked per year.

On further inspection of the data from 2006, we can look at the distribution of questions asked per month (see Figure 3.2). What is immediately noticeable is the absence of questions during the months of August, September and October. Due to a lack of concrete information⁴ to account for this, we can only speculate that there was a glitch or similar issue at Yahoo! which resulted in the exclusion of this data. In addition, July and December both have question frequencies that are far below the other months, indicating that perhaps the data is slightly unreliable in terms of its completeness.

According to the dataset, most questions were asked in November, specifically around 800,000 questions. Within this month, the most popular day was the 28th November which saw around 60,000 questions, with 21:00 being the

⁴Correspondence with Yahoo! presented no further details or information regarding this gap in the dataset.

most popular hour during which some 3,500 questions were asked. Typically, users only asked a single question during this time, however, a small proportion (399/2778) did ask two or more questions. Users typically generated only a single best answer, but again a small set (220/2428) of users were credited for two or more best answers during this hour. Interesting, both July and December in 2006 are missing the majority of the monthly question data.

As 2005 has only a small proportion of the questions and may include artefacts relating to the launch of Yahoo! Answers, it seems sensible to only consider questions from 2006, where the Q&A service is running more consistently and is in active use.

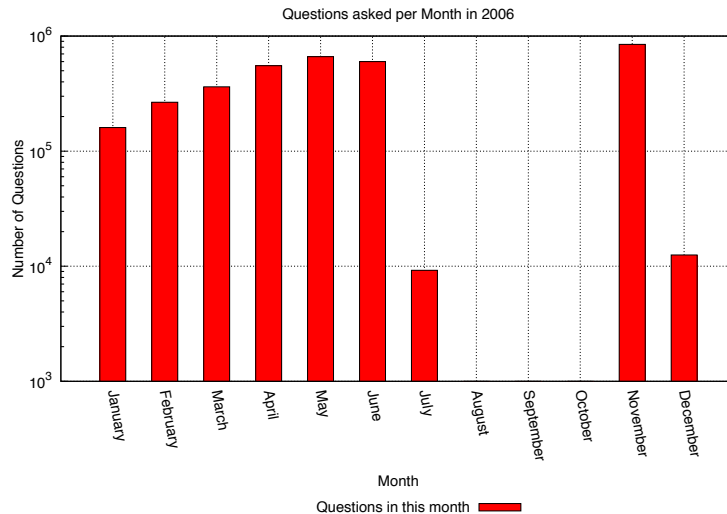


Figure 3.2: Yahoo! Answers questions per month in 2006.

3.2.4 Corpus Issues and Discrepancies

The users from the Yahoo! Answers dataset have potentially been using the service for up to one year.

Unfortunately, the dataset fails to link individual answers back to the user who composed them if they were not classified best answers. The dataset includes a vast quantity of unique answers to questions as seen in Figure 3.3, from which we can only assume that typical users will provide answers in more categories than they provide *best answers*.

The dates of completion and resolution in several entries are incorrect based on the definition and rules of the system. Following correspondence with a member of Yahoo! staff it was accepted that glitches or errors in batch procedures may have taken place. Fortunately, these particular glitches will not have an impact on the features of interest in this study.

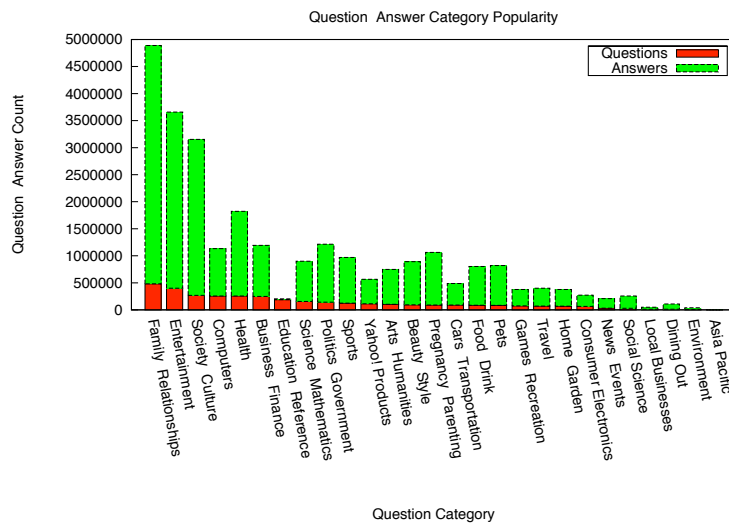


Figure 3.3: Corpus Q&A counts.

3.2.5 Category Popularity

Certain categories within the dataset are more popular than others, which can be seen in Figure 3.3 as a stacked histogram, and more graphically in Figures 3.5, 3.6 and in Table 3.7. Popular question categories are almost always more popular with answers (see Figure 3.3). It is clear that many more answers than questions exist, which is an artefact of Yahoo! Answers permitting multiple answers to a single question (as illustrated in Figure 3.4). Each question typically receives less than 100 answers, yet 100 answers represents a huge amount of effort and attention for a single question, arguably 10 or 20 answers could be considered excessive for a single question and may result in wasted time and effort. Of particular interest is the rise at 100 answers, this is due to “featured” questions which are presented prominently to encourage answers⁵,

⁵http://answers.yahoo.com/question/index;_ylt=AtxgLUteNPLZ58VB7wiZFR0jzKIX;_ylv=3?qid=20060622135529AAols2u

demonstrating the major issues associated with flooding a question to the entire community of answerers. Featured questions are thought to be selected from Yahoo! Answers editor's personal selections⁶.

From this dataset we can extract the distribution of category popularity for asking and answering questions, as well as the range of different categories users are most likely to contribute towards. This provides a useful and detailed insight into a real population of users' interests and expertise.

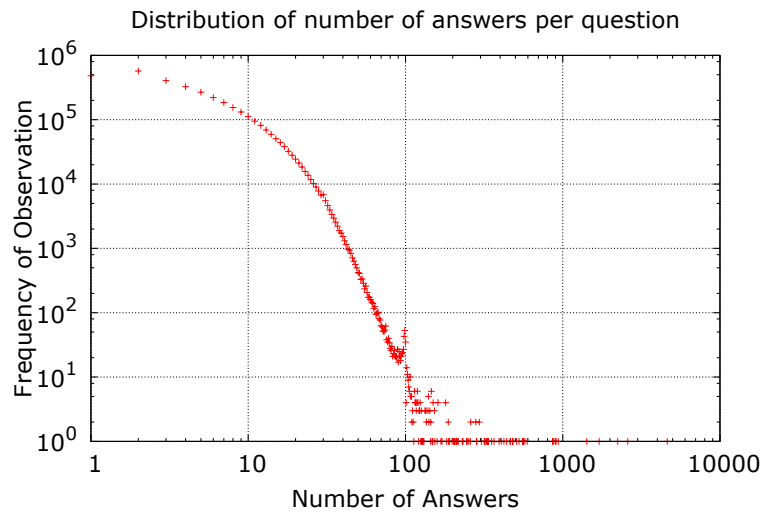


Figure 3.4: Number of answers per question.

3.2.6 Interests and Expertise

The dataset contains anonymized user identities in the form of a number. Each question and best answer relates back to a user identity. This allows for the analysis of the number of questions asked per user, the range of categories in which they were asked, the number of best answers supplied and the range of categories in which they appeared. In the context of this work, interests and expertise are defined as follows:

- **Interests:** a users' 'interests' are determined by the distinct range of top level categories in which they post questions.

⁶<http://answers.yahoo.com/question/index?qid=1005121503952>

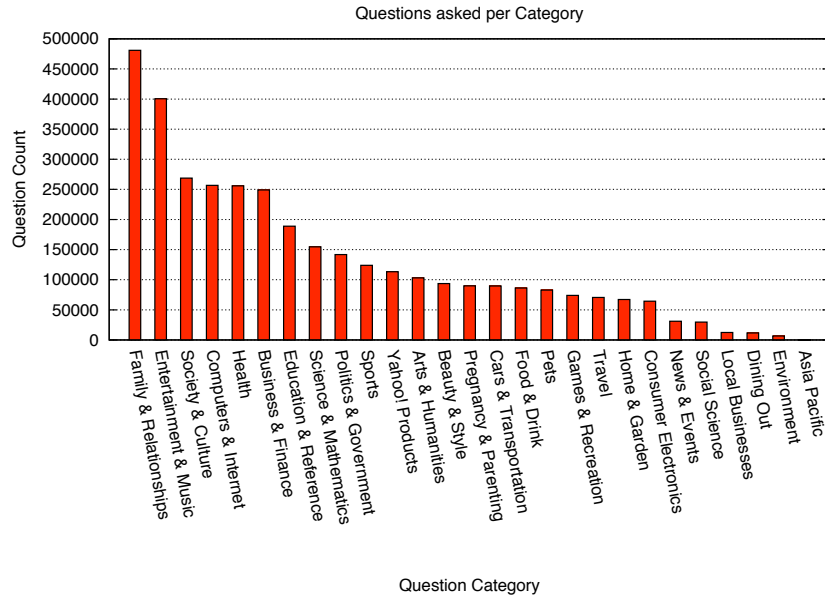


Figure 3.5: Question category popularity.

- **Expertise:** the distinct collection of categories in which a user has best answers is referred to as their ‘expertise’ category collection.

By analysing the range of interests and expertise of the user population we gain some understanding of what expertise are available. By taking the distinct set of unique question categories each user made contributions in, we can create a distribution of the various values. As previously highlighted, unfortunately we do not know directly from the dataset which answers each user gave, but instead only the answers which have been voted as being a best answer. We can assume that users will have a greater number of categories in which they will answer rather than simply those categories in which they gained best answers.

By examining Figure 3.8 which looks at the distribution of the number of interest and expertise categories, we can see that many users have multiple interests, and overall more have interests than expertise. This distribution shows that more users are more likely to have a broader range of expertise than interests at the higher end, while having a smaller range but a greater number of interests at the lower end.

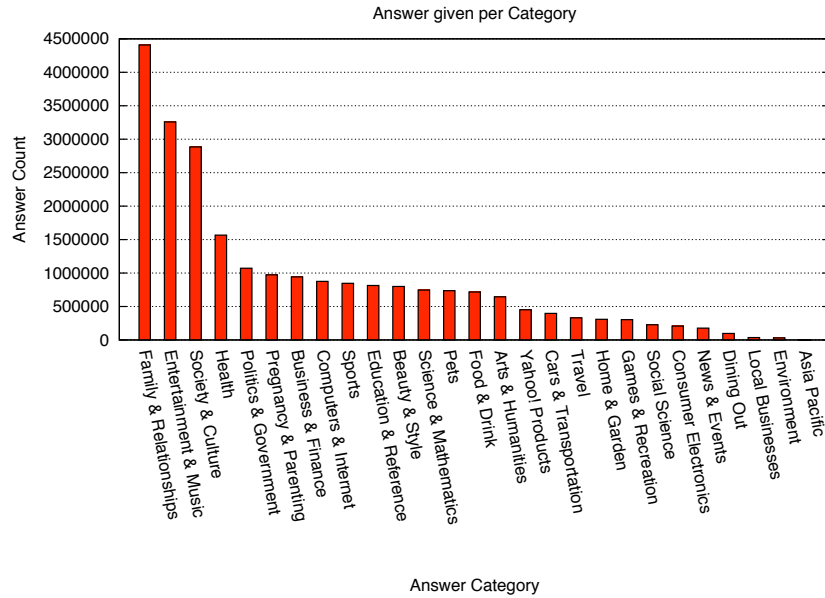


Figure 3.6: Answer category popularity.

In terms of modelling a user, we can draw from the discrete data distributions a number of unique categories a user will probabilistically ask and answer questions in.

3.2.7 Expertise Levels

By evaluating the quantity of best answers provided by each unique user in the dataset we are able to present an overall picture of the levels of expertise within the Yahoo! Answers community.

Figure 3.9 shows the distributions of how many best answers each user in the dataset has generated. The majority of users have zero or very few best answers, a small number of users have a handful of best answers, then there is a small number of users who collectively hold the highest number of best answers. Still, the analysis shows that a large proportion of the dataset users have a best answer in at least one topic. Furthermore, this distribution of expertise follows suit in all top level Yahoo! categories (see Appendix 8 for confirmation).

It can and has been argued, for example by Gyongyi et al. [38], that best answers do not paint an exact picture of expertise due to users voting incor-

#	Category	Questions	Answers
1	Family & Relationships	480883	4408408
2	Entertainment & Music	400620	3257625
3	Society & Culture	268765	2884617
4	Computers & Internet	256737	874911
5	Health	255829	1566000
6	Business & Finance	249163	942926
7	Education & Reference	188898	814794
8	Science & Mathematics	154550	745439
9	Politics & Government	141859	1071384
10	Sports	123991	846002
11	Yahoo! Products	113353	451175
12	Arts & Humanities	103130	645008
13	Beauty & Style	93678	798771
14	Pregnancy & Parenting	89874	972419
15	Cars & Transportation	89624	397274
16	Food & Drink	86296	716926
17	Pets	82875	737256
18	Games & Recreation	74057	302468
19	Travel	70539	329622
20	Home & Garden	67121	309469
21	Consumer Electronics	64361	206947
22	News & Events	31158	177471
23	Social Science	29615	227183
24	Local Businesses	12499	36551
25	Dining Out	11683	96746
26	Environment	6582	31502
27	Asia Pacific	254	1305

Figure 3.7: Category popularity within the Yahoo! Answers dataset.

rectly or maliciously. However, it seems appropriate to present a view of the levels of expertise within the dataset via this volume of best answers metric. Answers generated by a user with a large number of best answers are likely to provide accurate, helpful and useful content to user generated questions. An interesting extension (which lies outside of the scope of this thesis) would be the expanded evaluation of best answers using techniques such as those discussed and used by Gyongyi et al. [38].

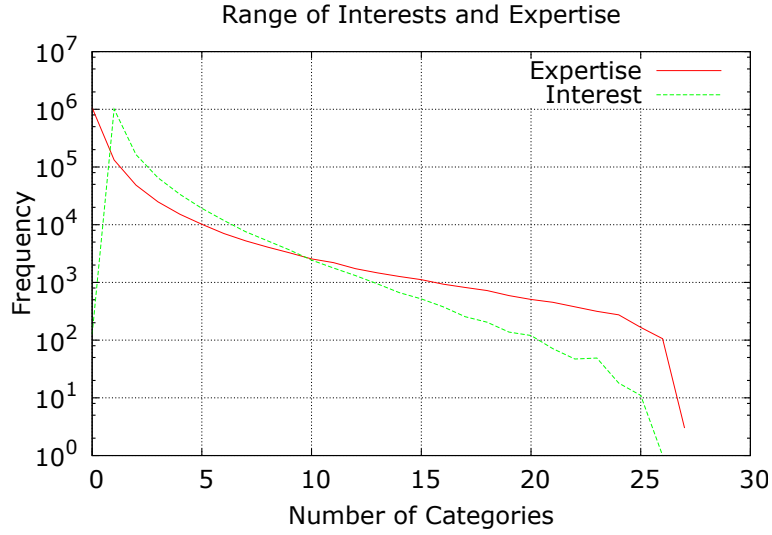


Figure 3.8: Number of interest and expertise categories per user.

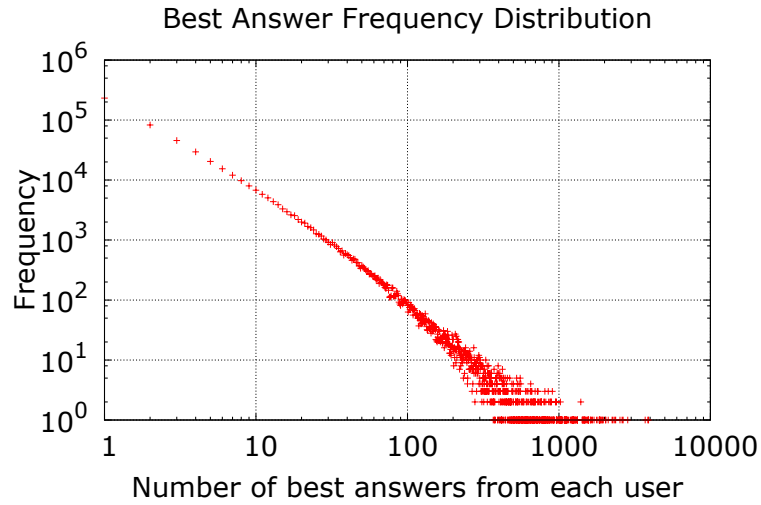


Figure 3.9: Number of best answer distribution.

3.2.8 Question and Answer Lengths

Questions consist of a 'subject' and an optional 'content' while answers exist simply as a single body of text. Question lengths are considered as the sum of the length of the subject and any additional content. The maximum question subject length is 110 characters. The maximum question content length is 1000

characters. A small proportion of questions contained longer lengths, but this was due to special privileges given to a small number of users because of higher ratings or scores within Yahoo! Answers. These questions have been ignored within the analysis as they are unrepresentative of the mass population.

The length of questions and answers in words is recorded, as presented in Figure 3.10. The majority are short and contain only a few words but there is no limit on this length and the Yahoo! Answers service does support questions and answers of considerable size, such as those in excess of 1000 words. This is perhaps due to the possibility of composing text over time without the any requirement for the author to remain constantly online while answers are composed.

Q&A length distributions can be used to model various question and answers generated by simulated users. Texts of various length take different amounts of time to read and compose and consequently different amounts of time to process.

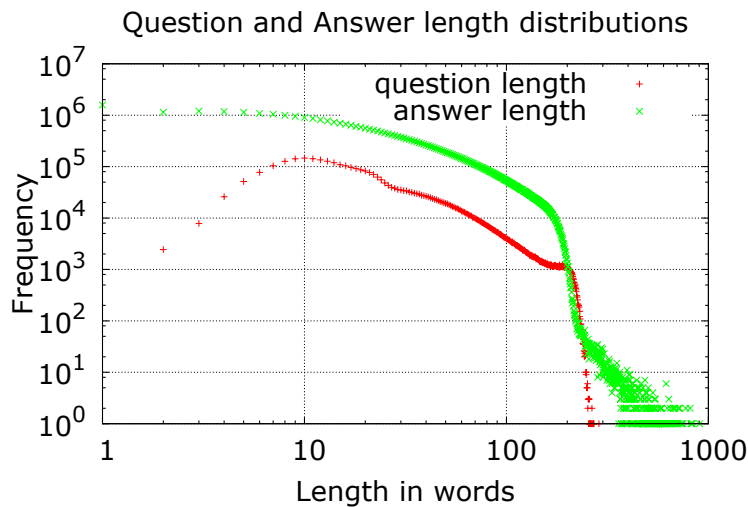


Figure 3.10: Q&A length distribution.

3.3 Twitter Corpus

To gather more information on user activity patterns in online communities, an experiment took place using information and status updates from Twitter

users. This took place for several weeks during 2010, after the specifics were approved by the University of Sussex Research Ethics Committee⁷. The content of tweets and user identities are not really relevant to this study, instead the aim was to investigate a population of users to find patterns which might be useful for user modelling. Twitter is an application where a significant proportion of use is mobile [28] and is doing extremely well. It is useful for users on-the-go and gives an image of user activity and behaviour patterns. Twitter has many users and is currently a popular pastime activity which provides accessible data to examine user patterns. Tweets consist of a text update up to 140 characters in length.

3.3.1 Twitter Application Programming Interface

The Twitter data can be accessed through an API via a valid user account. After testing the possibilities available for taking advantage of this API, a popular, widely used and supported Java library, Twitter4J⁸, was chosen. Twitter4J provides wrappers for making remote calls to Twitter to request and submit data via its API.

A brief overview of the Twitter API follows. A valid Twitter user may request the profile of a particular Twitter user via their username or user ID. A user's profile typically consists of their name, an image, a short sentence describing the user, their location and an external website link. Each user has a set of followers (the users who subscribe to their updates) and a list of friends (the users they themselves are following). The API supports requests to gain lists of a particular user's friends and followers. The collection of tweets from a particular user can also be requested. Each tweet includes the text update made and a time stamp of when it was created.

3.3.2 Bespoke Twitter Crawler

The Twitter API enforces rate limiting. The number of requests per user per hour is currently limited⁹ to 150 per IP address and 350 with a registered account. Official application developers can become 'whitelisted', increasing rate limits into the thousands, however, for research and academic purposes the standard limits apply. These limitations cause problems for harvesting

⁷<http://www.sussex.ac.uk/res/1-6-12.html>

⁸<http://twitter4j.org>

⁹<http://dev.twitter.com/pages/rate-limiting>

large amounts of data, so a bespoke crawler was created using a handful of unique hosts and accounts.

A MySQL database table was used as a priority-based queue, accessible from several machines operating the same program code. Every few minutes a time-based job scheduler (cron) job was triggered on each machine, whereby the most important priority task is requested from the database and then processed and executed. The MySQL *"FOR UPDATE"*¹⁰ keyword was used to ensure locking of table cells while executing a particular job in the queue. This gave the ability to select the highest priority task without interfering with other hosts using the same database table queue. If a problem occurred while executing a particular task, the table cell containing the job details could be unlocked and made available for processing later. See Figure 3.11 for a high level diagrammatic view of the crawler.

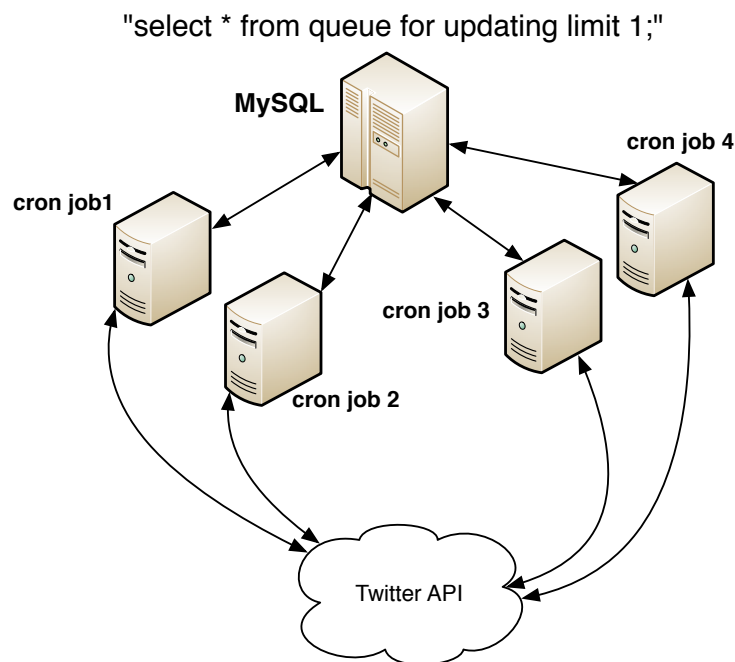


Figure 3.11: Bespoke crawler setup.

It is difficult not to be biased when sampling some set of users on Twitter, perhaps hooking into a set of automated commercial programs, extremely ac-

¹⁰<http://dev.mysql.com/doc/refman/5.0/en/innodb-locking-reads.html>

tive users or users who show strange anomalies in activity patterns which may skew findings. After careful consideration, a specific university-related account was chosen as an initial seed because it was official and would likely attract genuine interest from real users affiliated with the university.

Ultimately, the twitter corpus is a collection of users and their tweets based on a random walk with a bias towards those users who communicate within the community. The sampling technique used here is similar to that of *snowball* sampling¹¹, where existing subjects of interests recruit their acquaintances. This technique is thought to be like ‘rolling a snowball’. Such a sampling technique is often thought impossible to make unbiased samples impossible. For example certain people with a greater number of friends are more likely to be included in the sample.

When a particular user’s tweets were downloaded, the following rules were applied to determine which user should be crawled next:

1. Communicating users were downloaded with a high priority, based on @ sign usage found within tweets.
2. Connections of users who have been communicating directly via @ sign usage with a slightly lower priority were downloaded.
3. Other users and connections we came across with a lower priority were downloaded.

The crawler was capable of performing three particular functions using the Twitter API, these were:

getUser

Requests and downloads the complete profile data available for a specific Twitter user.

getTweets

Collects the entire set of tweets available for a given user.

getConnections

Downloads the user IDs of all users connected to a specific user.

The end result was 10,460 unique user profiles, plus their tweets which totalled a download of 6,589,096 status updates.

¹¹http://en.wikipedia.org/wiki/Snowball_sampling

3.3.3 Timezones

Figure 3.12 presents a list of the declared timezones of Twitter users within the dataset (those referenced by at least 20 users). The majority of users opt out of providing this information, however, there is still a large proportion of users registered in London, the United States and Canada. Twitter also allows users to specify a more detailed location, however it is not useful for analysis here as it is not validated and can be anything a user desires. For example, several different combinations of words may be used to describe the same or similar locations (see Figure 3.13 and 3.14), and it does not account for factors such as spelling errors.

Location	User Count
null	2849
London	1756
Mountain Time (US & Canada)	1045
Eastern Time (US & Canada)	932
Pacific Time (US & Canada)	796
Central Time (US & Canada)	703
Quito	333
Hawaii	178
Alaska	120
Greenland	96
Berlin	60
Santiago	54
Sydney	47
Brasilia	43
Arizona	39
Paris	37
Tokyo	36
Edinburgh	34
Amsterdam	28
New Delhi	28
Brisbane	28
Rome	26
Madrid	22
Melbourne	22

Figure 3.12: Twitter users timezones.

Location	User Count
Brighton	200
Brighton, UK	91
Brighton UK	16
Brighton, United Kingdom	10
Brighton & Hove	9
Brighton and Hove	7
Brighton & Hove, UK	6
Brighton, England	6
Brighton, East Sussex	5
Brighton, East Sussex, UK	3

Figure 3.13: Twitter users around ‘Brighton’.

Location	User Count
London	271
London, UK	57
London, England	11
London UK	8
London, United Kingdom	6
West London	4
South London	3
Chiswick, London	2
London/Brighton	2
London Town	2

Figure 3.14: Twitter users around ‘London’.

3.3.4 Number of Active Days

Users within the dataset are active at various times and days by posting a tweet update in response to the question ‘*What are you doing?*’ [28]. Figure 3.15 shows the distribution of how many distinct days within the defined period of the dataset each user is active (grouped within 25 day range bins). The majority of users are only active on a few days, but there are of course users who are active far more frequently. This trend is identified in existing work which classifies Twitter users according to the number of updates they make and the extent of their usage, showing that the majority of tweets arise from just a small proportion of the Twitter community, for example; celebrities and

organisations¹².

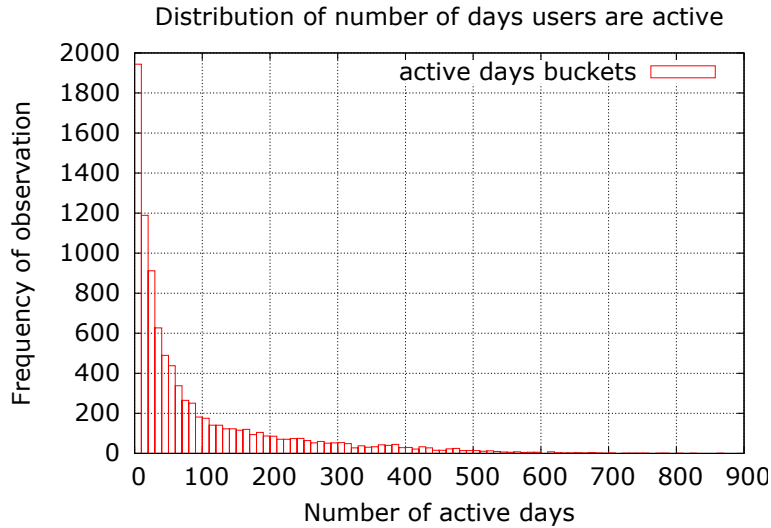


Figure 3.15: Distribution of days users are active.

The following figures include a variety of radically different user behaviours. Some users are using the twitter network for business or only for personal needs. The distributions of the number of tweets each user has submitted and the number of days each user is active for vary greatly (Figures 3.16 and 3.15). As such those users who are the most activate in the network have the largest influence of the statistics presented here.

3.3.5 Update Frequencies

Across the dataset, users produce various volumes of unique tweets. This distribution can be seen in Figure 3.16. The majority of users in the dataset have few tweets, however, there is a proportion of very active users who have in excess of 1000 tweets.

3.3.6 Inter-Tweet (Repeat Activity Times)

The inter-tweet time is the time between consecutive tweets by the same user. If a user tweets at 12:00 and then again at 13:00, the inter-tweet time is exactly one hour. Twitter users in this study are likely to tweet sooner (rather than

¹²<http://www.pearanalytics.com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009.pdf>

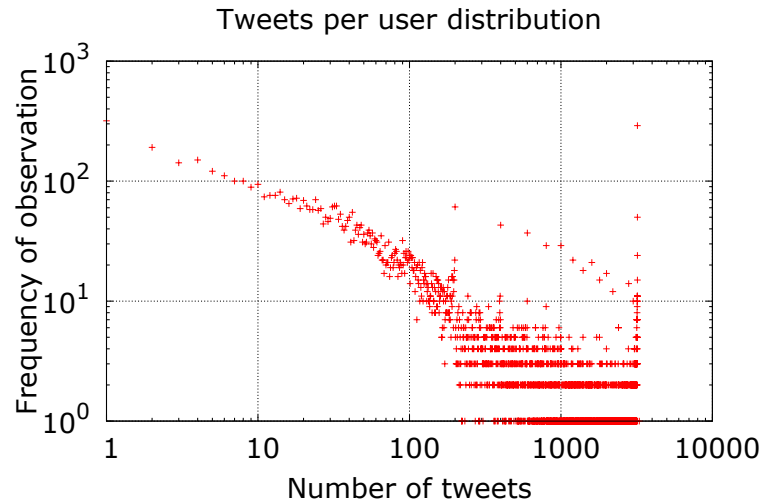


Figure 3.16: Distribution of tweets per user.

later) after their initial tweet. This can be seen in Figure 3.17, where the majority of recorded inter-tweet times are very low. Periodicity peaks around the 1,440 minute (1 day) level show that some users show routine in their tweeting habits.

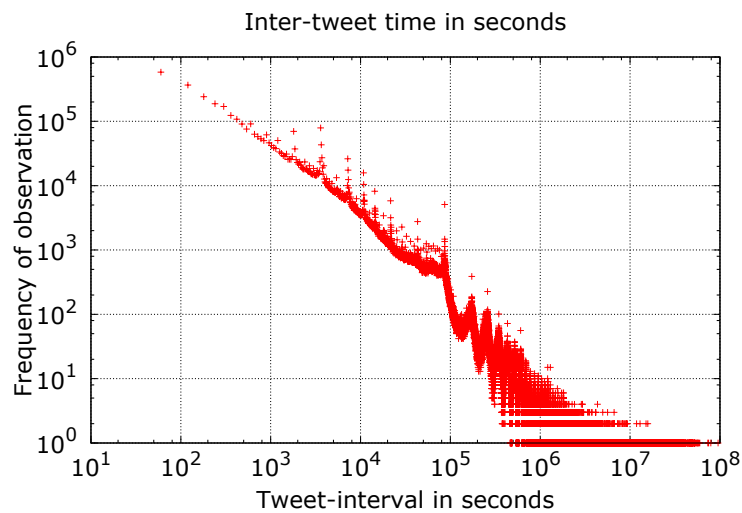


Figure 3.17: Inter-tweet time distribution.

3.3.7 Inter-tweet Reply Time (Response Times)

Many of the tweets analysed in this study are made as a direct response or reply to an existing tweet. The analysis shows that approximately 1,000,000 out of 6,589,096 tweets are a response, however, only 13,000 tweets from the dataset are a response to another tweet found within the dataset, as the original tweet may have been too far in the past or from a user whose information was not collected. Figure 3.18 shows the distribution of the interval between the posting of an original tweet and a response (an inter-tweet reply time). Although this study looks at a relatively small sample of data, in terms of Twitter as a whole it is clear that the majority of users are likely to respond quickly to a tweet, often within just a few minutes.

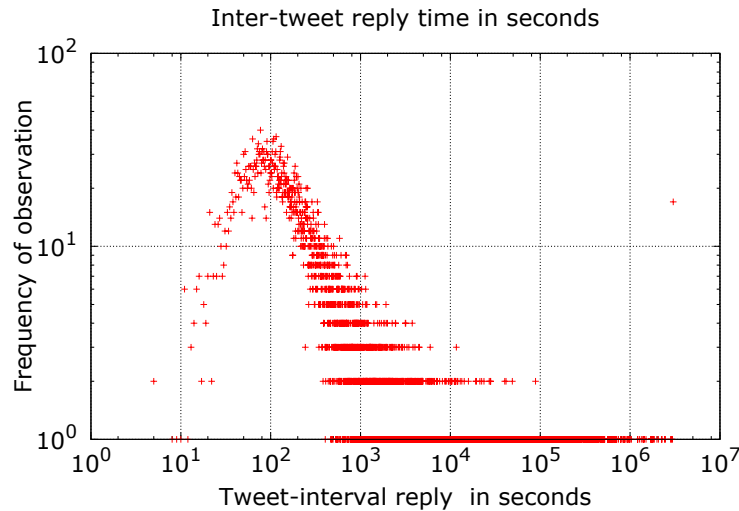


Figure 3.18: Inter-tweet reply time distribution.

3.3.8 Inter-Tweet Question Time (Question Intervals)

It is possible to perform a trivial analysis of tweet content to see if a question is being posed by a user, in which case encouraging a more prompt response or answer than it would if it was simply a random comment. The interval between asking questions can be seen in Figure 3.19, whereby tweets are considered to be a question only if they end in a question mark. Although a trivial method, which does not account for rhetorical questions or mistakes, it provides some indication of the volume of questions within the dataset. Again, it appears

that users are more likely to have short intervals between asking questions rather than long periods between asking multiple questions.

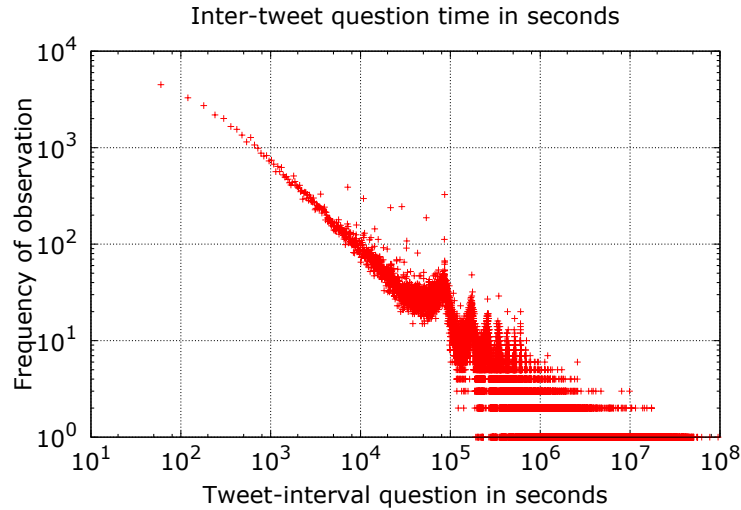


Figure 3.19: Inter-tweet question time distribution.

3.3.9 Tweet Replies (Popularity)

It is difficult to determine which tweets are made as a response when they are not directly flagged via the API user interface monitors (`reply_id`). The @ addressee tag can be used as a means to match up recent updates where it is likely that the tweets are made as a response to whoever the name @ is addressed to. Similarly, content analysis can be used to determine the link between related tweets, though this would be an extremely time-consuming task when carried out on a large scale. Fortunately, some tweets which were identified as being a direct response to a given tweet are identifiable directly via the Twitter API. From those tweets contained within the dataset, we can analyse the distribution of how many unique tweets match with a specific tweet ID. As shown in Figure 3.20, the majority of tweets which receive a response often only get a single reply. Typically however, most tweets are unlikely to gain a single response. This is almost certainly related to the number of followers a given users has, or to their importance within the Twitter community. As an example, the maximum number of replies found within

the dataset is to a tweet originating from someone from the Jackson 5 which received 126 responses¹³.

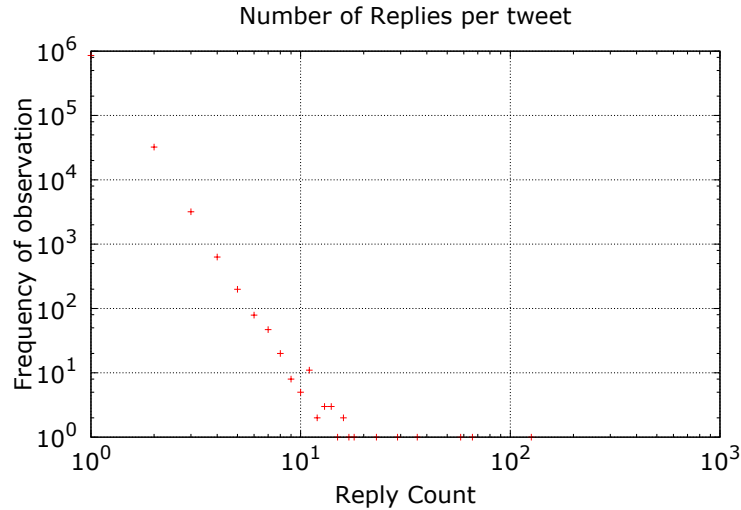


Figure 3.20: Number of replies to a tweet distribution.

3.3.10 Conversation Lengths (Chains of Interaction)

In a similar manner to investigating the number of direct replies a tweet has, we can also examine the *chains* of interactivity. From a given tweet not classified as a reply itself (the possible root of a chain), we can look at the direct responses it receives, as well as any further responses to those replies. The conversation length can be thought of as a chain of communication amongst Twitter users. As with all of the Twitter data analysis, it is only possible to investigate the chains of communication from those tweets *within* the dataset, so the users who are communicating and the tweets that *exist* between them have been analysed. The distribution of conversation lengths for this study can be seen in Figure 3.21. Conversations are typically short, with the odd longer exception emerging.

¹³<https://twitter.com/#!/titojackson5/status/2183192106766336>

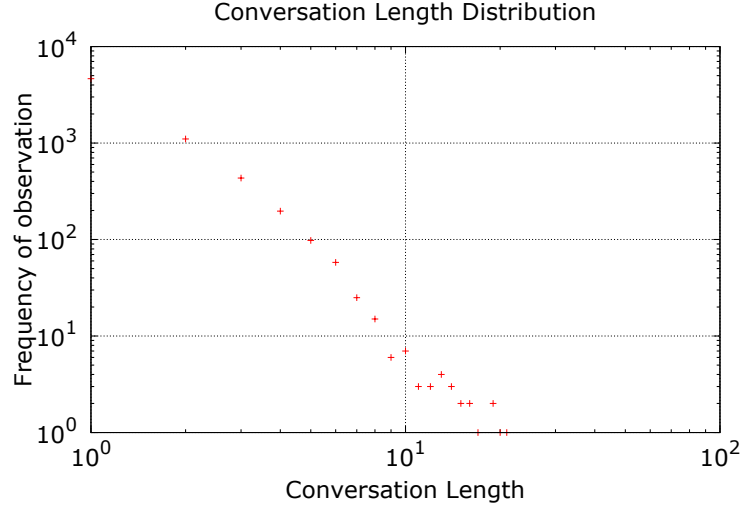


Figure 3.21: Conversation length distribution.

3.4 User Task Processing

In the context of this work, the users of a system can be thought of as processing units who generate questions and answers with various levels of competency. A question may be considered as a job or task which requires processing, then in response, a result in the form of an answer is created. We can consider this the high level abstraction of the processing tasks taking place in Q&A systems.

A key piece of research into human dynamics by Barabási [75] has been recently conducted, in which the author challenges the notion that human actions are randomly distributed in time and well-approximated by Poisson processes [75]. They address the *increasing* evidence that many human patterns – including *communication* – follow non-Poisson statistics, which can be characterised by bursts of rapidly occurring events, separated by long periods of inactivity. The authors show that this “bursty” nature within human behaviour is a consequence of a *decision-based queuing process*. Most tasks are executed rapidly, while few experience very long waiting times, which explains the heavy tails observed during their study. An experiment is presented with current models of human activity based on Poisson processes, assuming that within a dt time interval, an agent (human) focuses on a specific action with probability qdt , where q is the observed overall frequency of the activity.

Such models assume that the time interval between two consecutive actions by the same agent (the waiting time or inter-event time), follows an exponential distribution. The authors state that this interval is better approximated by a heavy-tailed or Pareto distribution: a dataset of e-mail logs is studied to support this case.

The priority decision queue is ordered by the task priority and executes in highest-priority order. Within a Q&A system, the priority could be computed as a function of how *interesting* the question is, how *long it will take to address* and *ability to answer*, or some other suitable metric.

The authors state that “*once in front of a computer, an individual will reply immediately to a high-priority message, while placing less urgent or more difficult ones on its priority list to complete with other non e-mail related activities*”. The authors also state that “*most e-mails are either deleted right away (which is one kind of task execution) or immediately replied to*”. Perhaps this is similar to answering or forwarding a particular message. Most interestingly the authors state that “*only the more difficult or time-consuming tasks will queue on the priority list*”, but also that “*the priority of a response is more important than the message size*”. It would seem that a priority-based queue using a suitable ordering metric is a sensible means to represent the task or job list for Q&A users.

Another relevant work here is that by Kleinberg [76], whereby a Markovian model is used with two states q_0 and q_1 . While in state q_0 messages intervals are set *low*, while in q_1 a *higher* interval is used. Such a mechanism is used to create bursty periods. A state transition takes place with probability $p \in (0, 1)$, remaining in the current state with probability $1 - p$, independently of past state transitions.

The priority-based queue method will be used in this research to represent a user’s current task list, and ordered by the interest level of questions.

3.4.1 User Attention

A user can be thought of as paying or consuming attention while performing some computer-orientated task, a simple example being to create a program in a programming language. Attention is consumed while composing the program and evaluating the outputs. The attention of a user can be monitored within the context of this work by noting the levels of input spent on a given task. The input level may indicate the complexity of the task, however, the time

taken would also seem an appropriate metric to allow for comparisons to take place.

In an ideal situation, when part of human-orientated services, users would be thought of as endless resources who continually process jobs within the system. In a more realistic scenario, users will spend some proportion of their time actively participating in the system and some proportion performing other tasks.

3.4.2 User States

A user can be thought of as being in a particular state from a given set of possible states. This collection of states and the transitions between them can be represented as a Markovian model, whereby the next state of a user is determined solely by their current state. Users may transition between paying attention and being away from the system, and also actively performing some human computable operation, such as answering a question. Such a state model can be seen in Figure 3.22, where attention can be consumed by a user while asking and answering questions in the attention state. Various fixed probabilities can be used to transition between states in this model. The proportion of time users spend paying attention and being idle is an interesting variable to consider within real-time Q&A. Users will remain in the attention state with probability P and will transition and return to idle with probability $1 - P$. Users will remain idle with probability Q and transition to the attention state with probability $1 - Q$. While paying attention users may be *asking*, *answering* or *waiting*. While idle the node software continues to route questions and answers. This model provides the flexibility to represent various attention behaviours.

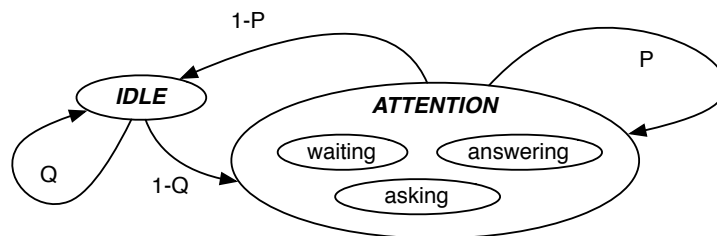


Figure 3.22: Markovian user state model.

3.4.3 Composing and Reading

Users will read and write questions and answers using various levels of competency. From existing studies into user proficiency, it is clear that users will require some period of time, effort and attention to compose questions and their answers. The most widely used text entry performance metric is words per minute (WPM) (see Equation 3.1) [77]. One study investigated the average number of WPM for keyboard and mouse entry in comparison to speech recognition techniques, looking specifically at *composition*, where for example, users articulate the generated text [78]. An average rate of 19 words per minute was achieved using a standard keyboard entry method.

Existing studies provide experiments which prove that the particular text entry method used will have an effect on the speed at which a user might be able to compose text. In addition, the level of expertise that the user has with the particular entry method will improve the rate at which they can input text into a system, whereby expert users may enter text significantly faster than average users [79, 80]. Finally, the amount of exposure to a particular text entry method also determines the speed at which a user enters text.

A second device specific metric of interest is Keystrokes per Character (KSPC) (see Equation 3.2), which defines the ratio between the length of the input stream to the generated text. This can be thought of as the amount of effort required when using a particular input device.

$$\text{WPM} = \frac{|T| - 1}{S} \times 60 \times \frac{1}{5}^{14} \quad (3.1)$$

$$\text{KSPC} = \frac{|\text{Input Stream}|}{|\text{Text}|} \quad (3.2)$$

Again, it is clear that the particular technology used for reading text determines the number of WPM a user may achieve [81]. Key features of a display include its resolution in Dots Per Inch (DPI) and physical dimensions, and experiments have shown that typically we read computer displays more slowly than printer paper, and that subjectively higher resolutions appear to have sharper character contours.

¹⁴Where s is the period between the first and last keystroke, and one fifth is the average length of a word. T is the text and $|T|$ is the length of T .

Concrete WPM reading and writing measures can be used within simulations to represent the time taken and attention consumed for users to read and write content generated within the network. Experiments can be used to understand and dictate the levels of text entry and speed of reading that would be required for Q&A networks to flourish.

One study provides text entry WPM averages attained during a study for various input devices [77]. The values, see Table 3.2, can be used to model various input environments of users.

Method	WPM	SD
Physical (QWERTY)	75.84	15.61
Projection (QWERTY)	46.60	-
mini-QWERTY	50.86	15.68
Stylus-Based	11.62	3.37
Soft / Virtual Stylus	24.88	7.78
Twiddler	31.75	7.85
Standard 12-Key	9.94	2.72

Table 3.2: Average WPM for text entry methods.

3.5 User Churn Models

Users will arrive and leave a given system at different rates and times. This user behaviour is known as churn, and is particularly significant in peer-to-peer networks. Much research on network churn investigates churn levels in relation to the percentages *churning*, for example, as explored in work by Jelasity and Montresor [82], can a system cope when $X\%$ of the population is randomly removed and replaced at set intervals?

Random replacement of users is an effective means to explore how a system reacts to churn and how robust a network is, but it could be more realistic. With the rise of mobile devices capable of initiating network connections at any time, patterns and options for churn levels should be considered in more detail. It is possible to consider several other churn models to create a more realistic pattern of networked user behaviour based on real user activity patterns, this analysis follows.

3.5.1 The Weibull Distribution

Several significant papers investigating electronic churn patterns have come to the conclusion that it matches up most closely with the Weibull distribution [83, 84]. Past research often considered churn levels to follow an exponential distribution, however, more recent analysis has proved otherwise. We can assume that Q&A users will behave in a similar manner to those using P2P systems. The Weibull distribution was first named after Waloddi Weibull in 1951, but first devised by Fréchet in 1927 and then implemented by Rosin & Rammner in 1933.

It is based on two parameters: a shape parameter k and a scale parameter λ . It is related to several other distributions, in which it interpolates between the exponential (when $k = 1$) and Rayleigh (when $k = 2$) distributions. The Weibull distribution is therefore extremely dynamic and offers a means to model various and varied continuous distributions.

The probability density function of the Weibull distribution [85] can be calculated via equation 3.3.

$$f(x; \lambda, k) = \frac{kx^{k-1}}{\lambda^k} e^{-(x/\lambda)^k} \quad (3.3)$$

The cumulative distribution function of the Weibull distribution [85] can be calculated using the equation 3.4.

$$f(x; \lambda, k) = 1 - e^{-(x/\lambda)^k} \quad (3.4)$$

Works such as Stutzbach and Rejaie [83] and Xiao et al. [84] have clearly identified the Weibull distribution as matching the churn levels found in real computer networks.

Stutzbach and Rejaie [83] conducted an investigation into churn found in three popular P2P systems: Gnutella, BitTorrent and Kad. They looked at suitable distributions to describe session lengths, and determined that the Weibull Distribution is the most similar. In this specific work, a Q&A system probably has most in common with an unstructured file sharing system such as Gnutella. Answers could be thought of as files which a user can create on an ad hoc basis to some level of proficiency, while questions can be thought of as search requests with associated parameters. Importantly, the authors

realise and state: “Towards this end, researchers and developers require an accurate model of churn in order to draw accurate conclusions about peer-to-peer systems”. Indeed, this work concludes that peer-to-peer session lengths are best fit by Weibull or log-normal distributions, not the exponential or Pareto distributions as was previously thought. The availability of individual peers exhibits a strong correlation across consecutive days, showing that users exhibit repetitive behaviour. In BitTorrent, peers frequently remain in the system long after their downloads complete.

Later in 2007, Xiao et al. [84] also accurately matched the Weibull distribution against instant messenger session durations, as well as several other characteristics of instant messenger networks, including the number of contacts. The Weibull distribution includes a selection of other distributions which allows it to model a variety of different distributions.

The *inter-arrival distribution* captures the pattern of when peers will arrive in a P2P system, while the *session-length distribution* identifies how long they stay in the system. By using both the *inter-arrival distribution* and *session-length distribution* from a Weibull distribution, it is possible to model churn.

We can specify a Weibull session duration distribution in a sentence such as:

“Eighty percent of users stay for at least X minutes, while fifty percent stay for at least Y minutes.”

From such a statement with two data points, it is possible to determine appropriate k and λ values such that the distribution preserves these requirements (by solving the simultaneous equations).

This approach to expressing a distribution is far more intuitive than simply stating some parameters, and in addition, allows us to create arguably realistic session duration levels for various scenarios.

We are interested in constant network sizes, and as such require a constant departure rate and replenishment of nodes. We can achieve the desired inter-arrival rate Ω by noting the following:

$$\Omega = \frac{\text{Network Size}}{\text{Mean Session Duration}}$$

Thus, we are required to replace the entire network of nodes after the Mean Session Duration. The mean of a Weibull distribution is calculated by:

$$E[X] = \lambda \Gamma(1 + \frac{1}{k})$$

Therefore, we can trivially calculate the required value of the scale parameter λ by:

$$\lambda = \frac{\text{mean session duration}}{\Gamma(1 + \frac{1}{k})}$$

3.5.2 Possible Churn Scenarios

As we need to evaluate our system using various levels of churn, we have decided on the following requirements and the corresponding Weibull distribution parameters. This collection of parameters is by no means exhaustive, but it should provide a fairly intuitive range of user behaviours in regards to churn, and the effect on the resulting session durations are shown in Figure 3.23 and 3.24.

name	churn description	k	λ
C0	all nodes stay for the full duration	—	—
C1	25% up to 3 hours, 50% between 3-5 hours	3.07	269.79
C2	25% up to 0.5 hours, 50% between 0.5-2 hours	1.13	89.97
C3	25% up to 5 hours, 50% between 5-7 hours	4.67	391.64

Figure 3.23: Possible churn scenarios.

3.6 Probabilistic Modelling

From a given Probability Density Function (PDF), the associated Cumulative Distribution Function (CDF) is found and can be used to draw values from the range using a random variable x . This modelling allows for the extraction of a set of values, creating a realistic representation from the distribution being used.

For *discrete distributions* as found in Yahoo Answers! and Twitter, we are able to create CDFs by tallying values or classes of interest present in the data, summing at each stage to create a cumulative total. Finally, the collection of cumulative values can be normalised to the [0.0 1.0] range, from which a

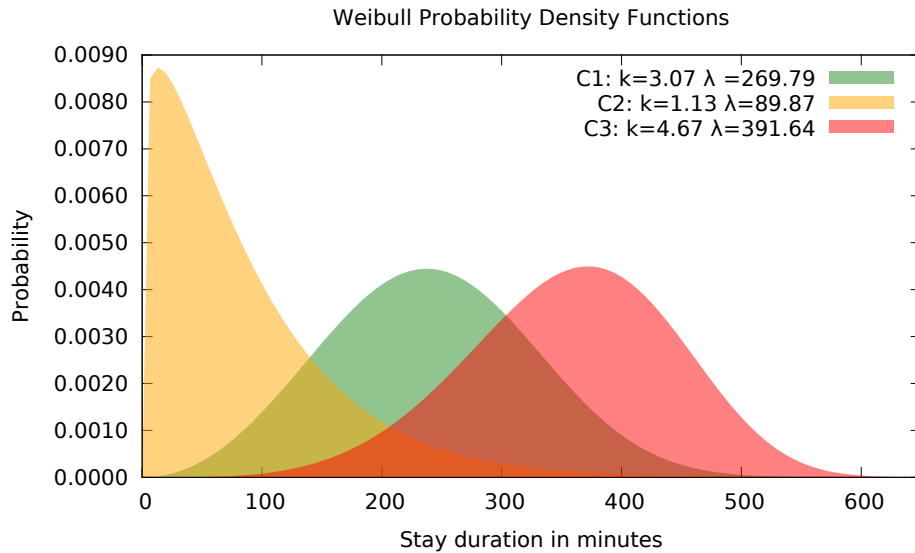


Figure 3.24: Possible churn scenario PDFs.

uniform random variable (such as those found in Java's `SecureRandom`¹⁵ Class) may be used to extract values from the distribution, whereby the probability of drawing a specific output is based on the likelihood that it occurred in the original real data. These discrete distributions are found by querying the SQL database, discussed in Section 3.2.1, to produce totals for the number of distinct users present in each class of interest in the various topics of interest e.g. the number of best answers per user.

3.7 Summary

This chapter has grouped together a collection of data analysis approaches to find key properties and attributes, along with population distributions, to aid the representation of typical Q&A users. Using probabilistic modelling and utilising random variables, realistic populations of Q&A users may be created for use within simulations. The essential elements of user modelling for the purpose of Q&A have been identified and discussed and the findings can be used to represent simulated users within communication networks to compare and contrast various approaches.

¹⁵<http://download.oracle.com/javase/1.4.2/docs/api/java/security/SecureRandom.html>

Deniable Routing for Q&A

This chapter presents several naïve approaches towards deniable Q&A as well as a new routing protocol which takes inspiration from stigmergy as seen in ants foraging for food. This routing approach aims to encourage questions to flow towards the members of the network who stand a better chance of answering well, while adhering to their privacy requirements by blending authors' identities into the crowd of networked individuals.

4.1 Homogeneous Network Topology

To support plausible deniability within the ad hoc decentralised model, single hop routing tactics are used to pass messages between network nodes. This feature of ad hoc networks allows privacy requirements to be supported by creating an element of anonymity or plausible deniability for question askers and answerers. This is in a similar fashion to work by Kacimi et al. [5] and Clarke et al. [6]. The complete path of the route of a particular question or answer is not known by any one node in the network, therefore the exact author of a specific question or answer is plausibly deniable.

In order for such mechanisms to work a uniform random topology is used for the network graph, promoting full connectivity by setting the degree at each node to Erdős and Rényi constant k [86]. Erdős and Rényi provide an equation for translating the size of a random graph ($\Gamma_{n,N}$) to a degree threshold across the graph that tends towards complete connectivity. Stating that: if $N > (\frac{1}{2} + \epsilon)n \log n$ where $\epsilon > 0$ then the probability of $\Gamma_{n,N}$ being connected tends to 1 if $n \rightarrow \infty$ where $\Gamma_{n,N}$ is some random graph with n vertices and N edges. N is treated as a required network edge threshold which is shared among the network by dividing the requirement across all network nodes.

4.2 Question Routing

When a question is injected into the network the asker generates a unique question Globally Unique Identifier (GUID), in the form of an Immutable Universally Unique Identifier (UUID)¹, and a Time To Live (TTL) value. GUIDs are used to uniquely identify questions so that routing choices can be recorded, allowing for path reconstruction later by relating a GUID to a particular neighbour. TTL values are used to prevent questions from lingering in the network indefinitely. At each hop (visited node) the TTL value is decremented and when the TTL value reaches zero the question is discarded. To prevent identification of the source of a question, we use a random Poisson distribution to assign our TTL values. Our random Poisson distribution has a mean value related to a proportion of the network size and the number of answers being asked as defined in equation 4.1.

$$poisson_mean = \frac{network_size * exploration_proportion}{answers_required} \quad (4.1)$$

The TTL values from equation 4.1 allow the exploration of a specific proportion of the network, divided between the number of answers requested.

4.2.1 Naïve Routing Approaches

This is the first attempt to evaluate routing tactics to aid deniable question answering within fully decentralised ad hoc networks, therefore intuitive routing control cases are used as opposed to comparisons with other approaches. In small networks, a flooding approach will perform well. However, flooding does not scale with the number of nodes and acceptable levels of attention. A random approach performs in all sizes of networks, however it neither learns, nor directs questions towards experts. In this section we describe the naïve approaches of network flooding and random hops.

4.2.1.1 Flooding

A *flooding* approach will attempt to deliver all questions to all network nodes. Each node that is able to answer the question will compose its answer and send it back towards the source node using single hop routing tactics. The

¹<http://download.oracle.com/javase/1.5.0/docs/api/java/util/UUID.html>

flooding approach should reach the best possible answerers but also the worst. In larger networks, the number of responses may be so large that they result in source bombardment or denial of service. It is also possible to push out less interesting questions from the experts due to the flooding of local priority queues with more interesting questions. In addition this approach consumes the maximum levels of attention from all users of the network. Figure 4.1 depicts the flooding routing tactic, where a single question is forwarded by each node to all its known neighbours. Once seen and forwarded, the node will ignore any subsequent occurrences of the same question.

4.2.1.2 Random Hops

A *random* technique will pass questions between nodes choosing an arbitrary path. This simple approach can find experts, but it is unable to differentiate between the levels and quality of answers and expertise. The main benefit is that it requires comparatively low overhead, but unfortunately, the answer quality is inconsistent and statistically worse than a more informed approach. The random routing approach can be seen in Figure 4.2. Each question is forwarded once randomly, with uniform probability and each link has an equal chance of being selected as the next hop up until it reaches its maximum number of hops (a function of network size).

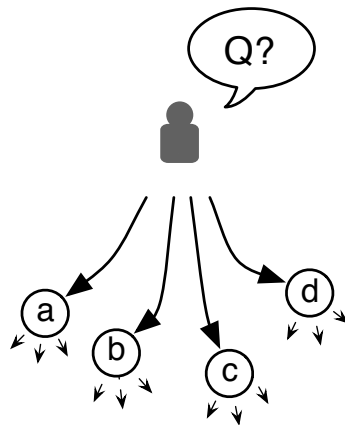


Figure 4.1: Flooding: send question to all links.

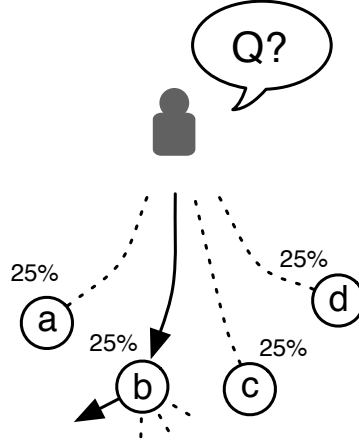


Figure 4.2: Random Hops: uniform path selection.

4.3 Stigmergic Routing

To locate food sources, ants will travel and explore away from the nest. As the ants travel they will deposit pheromones and those ants which are able to find a food source will doubly reinforce their own trail on their return. Strongly scented routes are more likely to encourage other ants to follow the same path. Over time, stronger scented routes appear towards the nearest food sources.

The aim of the routing in this work is to locate resources, however with the added complications of not wanting to over exploit a particular resource while preferentially selecting *better* sources. If we imagine the users of the network as sources of food of varying quality (which represents their ability to answer questions), we can use this technique to direct questions towards the more knowledgeable members of the network. Stigmergy does not require identities or state exchanges to take place between nodes and thus is particularly applicable to deniable routing.

Existing stigmergic routing research (see Chapter 2) aims to discover the best or least congested routes between hosts. We wish to locate any route to the best resources while ignoring reaching a particular host and the optimal path selection to reach them. As identified in existing research, multiple pheromone types will be used to represent various question contexts to aid path selection. The new use of stigmergy in question routing within Q&A

networks is presented in this section.

4.3.1 Overview of Technique

Pheromones induce trail-following behaviour in some ants, as discussed in Section 2.4. Real pheromones are chemical trails marking pathways towards a particular area of interest, for example, a food source. Pheromone scent levels increase as a function of the number of ants adopting a particular route (depositing pheromones) and disperse over time in an evaporation process. The more strongly scented a route is, the greater the probability that ants will follow that pathway. Within the context of communication networks, pheromones are represented as numeric values, signifying the probability of selecting a given route.

Taking inspiration from the features of stigmergic tactics as used by foraging ants, path selection can be made probabilistically based on *virtual pheromone* scent levels. Based on past interactions, the route of a question will tend to follow a path to a user or users with some proven competency in a given subject area. Where there are no past interactions to guide it, a random exploration process is used.

Pheromones can be used to reinforce the routes from which answers are generated for each question category, and in addition, positive feedback from question askers can be used to doubly reinforce those routes from which useful or better answers emerge.

Unlike most previous stigmergic approaches [7, 8, 9, 10, 11, 12, 13, 14] which proactively seek optimal solutions, only direct on-demand interactions within the network are needed to update the pheromone levels. Users cannot be expected to generate content solely to map the network, and one should not trust any suggestions simply presented by users without evidence in the form of explicit answers and feedback.

4.3.2 Question Routing Protocol

When a question arrives at a particular node, a routing choice needs to take place. The stigmergic protocol will forward questions to neighbours probabilistically, based on the pheromone strengths for the corresponding question category as presented in Figure 4.3. In this example, a user asks a question identified by a GUID, which is forwarded to node a , then to an arbitrary set

of intermediate nodes, finally arriving at node b who forwards the question to a node that is interested in answering.

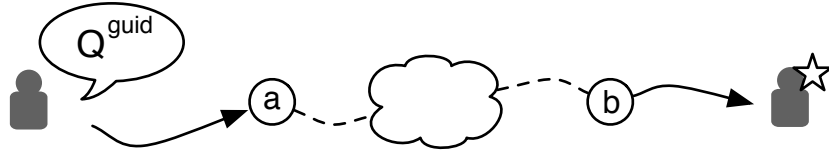


Figure 4.3: Protocol question message sequences.

No node in the sequence knows the full path taken by a question. A single node in the path between any question asker and answerer only knows which neighbour sent them the message and the neighbour they forwarded it to.

4.3.3 Answer Protocol

In time, a question is read by the answering node and a unique answer with a corresponding GUID' is sent back along the path from which the question originated (see Figure 4.4). An answer relates to its question via the original GUID to allow for path reconstruction back through the network. The answer causes local pheromone strengths (related to the question category) at each intermediate node in the path to be increased back towards the answerer (in the direction of the grey arrows beneath the nodes). This creates a greater probability of selecting this route in the future, when questions may happen upon one of the nodes in this pathway. In the example, a star icon indicates the answerer and represents their expertise in this particular question category.

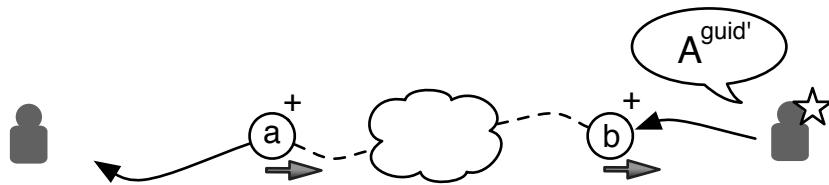


Figure 4.4: Protocol answer message sequences.

4.3.4 Feedback Protocol

The original question asker can send positive feedback to the corresponding answerer following the same route an answer took with GUID' (see Figure 4.5). Positive user feedback should cause a more powerful reinforcement of pheromones along the path towards the answerer, again increasing the probability of selecting this pathway for this particular question category in the future. It is assumed that all satisfactory answers receive positive feedback, but no further distinction between the accuracy level of answers is made. If the answer did not originate from a user with some expertise, then this feedback step would not occur and the pheromone level would only get the small boost associated with responding.

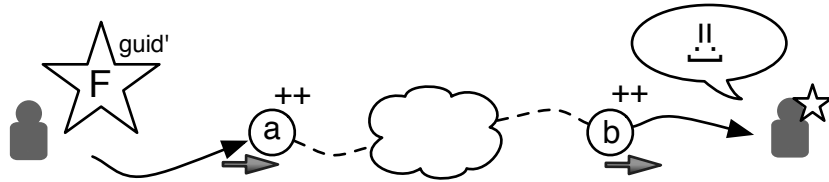


Figure 4.5: Protocol feedback message sequences.

The stigmergic routing approach seen here is based on and adapted from work by Barán [87]. This approach was able to perform well with unsupervised proactive routing and is therefore also applicable to the routing requirements here. The difference here is that the end user will choose to provide a positive backwards reinforcement rather than the protocol deciding automatically on path length or other performance metrics. The end user will decide if a feedback message is generated based on the utility of the answer received.

4.3.5 Routing Tables and Pheromones

A node's neighbours are described locally in a routing table. As in other peer-to-peer networks, the routing table contains a collection of addresses which are considered neighbours. In stigmergic approaches, a pheromone value is connected with each entry in the routing table, and in this case, for each possible question category type. For example, a neighbour will have N unique pheromone values for a known neighbour, where N is the number of possible

question categories that may be assigned to a question in the network. Figure 4.6 consists of three neighbours (A, B and C) and their associated pheromone values. It is worth noting that in any representation of this routing table, the category IDs would be used rather than full category name, but they are used here to make the information more accessible. The space requirements for this approach would be an N by M matrix, where N is the number of possible question categories and M is the number of current neighbours, aiming to be in line with the Erdős and Rényi constant k . In this thesis, the set of pheromones for a given link are commonly referred to as *category pheromones*. Therefore, the category pheromones of a node relates to the data represented in Figure 4.6.

In this thesis, offline nodes are automatically removed from local routing tables, which avoids the problem of forwarding to offline parties. This could be achieved however via the inherent mechanisms of the Transfer Control Protocol (TCP) or by using simple ‘keep-alive’ messages or acknowledgements, whereby unresponsive neighbours can be assumed to be offline. There is however a cost associated with requiring repetitive ‘keep-alive’ messages, which would not be exhibited by a less accurate approach such as connecting on demand, as and when required.

User behaviour could be cached to aid future bootstrapping, whereby a node would attempt to contact its previous neighbours when returning to the network before trying to contact new unknown nodes. This caching technique is left for future work as it is outside the scope of this thesis.

4.3.6 Pheromone Update Rules

When a particular event occurs it may directly update the pheromone levels at that given location.

The protocol messages flowing through the Q&A network are used in a similar manner to ants laying pheromones. The protocol messages will cause various quantities of different pheromones to be deposited on the routing table entries as and when interactions take place. The routing will not proactively seek to map the network or learn about its occupants while Q&A is not actively taking place, but instead will build and update the state of the routing tables during the exchange of messages. In this manner, the protocol simply sits on top of the interactions already taking place in the Q&A network.

#	Question Category	A	B	C
1	Family & Relationships	0.06	0.06	0.06
2	Entertainment & Music	0.06	0.9	0.06
3	Society & Culture	0.7	0.06	0.06
4	Computers & Internet	0.06	0.06	0.06
5	Health	0.06	0.06	0.15
6	Business & Finance	0.06	0.06	0.06
7	Education & Reference	0.06	0.06	0.06
8	Science & Mathematics	0.06	0.66	0.06
9	Politics & Government	0.06	0.06	0.06
10	Sports	0.06	0.06	0.06
11	Yahoo! Products	0.06	0.06	0.06
13	Arts & Humanities	0.06	0.06	0.06
12	Beauty & Style	0.06	0.06	0.06
14	Pregnancy & Parenting	0.11	0.06	0.06
15	Cars & Transportation	0.06	0.06	0.06
16	Food & Drink	0.06	0.06	0.06
17	Pets	0.06	0.06	0.06
18	Games & Recreation	0.06	0.06	0.06
19	Travel	0.06	0.75	0.06
20	Home & Garden	0.5	0.06	0.06
21	Consumer Electronics	0.06	0.2	0.06
22	News & Events	0.06	0.06	0.06
23	Social Science	0.06	0.06	0.06
24	Local Businesses	0.06	0.06	0.06
25	Dining Out	0.1	0.06	0.06
26	Environment	0.05	0.06	0.06
27	Asia Pacific	0.06	1.0	0.06

Figure 4.6: Local category pheromone values.

The following pheromone update rules can be used for updating routing table entries; i) increase strength to links which produce answers; ii) increase strength to those links which produce useful answers; iii) optionally reduce strengths to those links which have been forwarded questions recently (load balancing and a penalty for failure to answer without needing a timeout). These rules translate neatly to the three key messages found in the system, namely: *answers*, *user feedback* and *questions*.

This very small set of rules can be used to provide various outcomes, determined by the values used to update them. For example, if the update rule for

increasing the strength to those providing useful answers is low, then experts will be slow to gain preference in the network routing. On the other hand, if the strength increase is too large in comparison to the other update values the experts may be swamped by requests as inbound connections rapidly become increasingly appealing and exploration less likely.

After a number of experiments, update rules values were chosen (Figure 4.7) as a baseline since they perform well across a number of scenarios. Chapter 6 provides more detail to the selection of these values.

Variable	Value
Default Pheromone Value	0.06
Pheromone Minimum	0.01
Pheromone Maximum	5.0
Answer Increase	0.05
Feedback Increase	0.8
Load Balance Decrease	0.05
Evaporation Rate	0.0001
Evaporation Interval	250 steps

Figure 4.7: Local category pheromone values.

4.3.7 Probabilistic Path Selection

As per the previous section, pheromones are represented in the local routing tables of network nodes. T^i represents the routing table at node i , with each entry representing the learned appropriateness of choosing link l for questions of category type c , denoted T_{lc}^i . The appropriate routing entry values are used to make a probabilistic routing choice for questions at each node using a simple algorithm (see Equation 4.2), whereby the probability of path selection is directly proportional to the pheromone value for the given link or category combination within the local node's routing table.

The probability of selecting a given path is:

$$P_{lc} = \frac{(T_{lc}^i)}{\sum_{j=1}^n (T_{jc}^i)} \quad (4.2)$$

This equation is used within stigmergic routing via the use of algorithm 1 and 2 (see above). This allows for a probabilistic selection of the next hop for a question, based solely on the category pheromone levels found within local routing table entries.

Algorithm 1: SelectNextHop

Data: category
Result: The next hop for a question

```

1 total  $\leftarrow$  getPheromoneSum(category);
2 select  $\leftarrow$  total * nextRandom();
3 running  $\leftarrow$  0;
4 // Evaluate each node
5 foreach node  $\in$  neighbours do
6   | running  $\leftarrow$  running + pheromones[node][category];
7   | if select < running then
8   |   | return node;
9   | end
10 end
```

Algorithm 2: getPheromoneSum

Data: category
Result: The sum of pheromone for a given category across all neighbours

```

1 total  $\leftarrow$  0;
2 // Evaluate each node
3 foreach node  $\in$  neighbours do
4   | increment  $\leftarrow$  pheromones[node][category];
5   | total  $\leftarrow$  total + increment;
6 end
7 return total
```

The three key protocol messages; questions, answers and feedback, all relate to a specific category. From a given message we can extract the related category from associated metadata². When a particular protocol message arrives with a corresponding category, the routing tables can be dynamically updated as seen in algorithm 3. A node will increase the pheromone category strength for a neighbour (associated with the original question/answer routes) depending on the particular message received. We assign constant values to increase pheromone strengths for each message and also use a minimum and maximum pheromone strength constraint.

The `nextRandom()` call seen in Algorithm 1 is used to generate a random variable in the range of 0.0 – 1.0. The random variable is considered as a

²It might also be possible to extract this information using natural language processing techniques, but this is beyond the scope of this thesis.

Algorithm 3: IncomingMessage

Data: IncomingMessage
Result: Deal with all incoming messages

```

1 // Dealing with incoming Question messages
2 if type(IncomingMessage) == Question then
3   q ← IncomingMessage;
4   routing[qguid] ← ingressLink(q);
5   // Decrement time to live
6   qtll ← qtll-1;
7   // Is this a question category we wish to answer?
8   Boolean weCanAnswer ← areWeInterested(q);
9   if weCanAnswer then
10    // Add question to local priority queue
11    AddToLocalQueue(q);
12  else if qtll > 0 then
13    // Send question to a neighbour
14    Send(q, SelectNextHop(qcategory));
15  end
16 end
17 // Dealing with incoming Answer messages
18 if type(IncomingMessage) == Answer then
19   routing[aguid] ← ingressLink(a);
20   a ← IncomingMessage;
21   Boolean ourAnswer ← answerToOurQuestion(a);
22   category ← acategory;
23   // Increment pheromone levels towards answerer
24   node ← pheromones[node][category] + x;
25   if ourAnswer then
26     // Give answer to user
27     SendAnswerToUser(a);
28   else
29     // Send answer towards question asker
30     Send(a, routing.get(aquestionguid));
31   end
32 end
33 // Dealing with incoming Feedback messages
34 if type(IncomingMessage) == Feedback then
35   f ← IncomingMessage;
36   node ← routing[fanswerguid];
37   category ← fcategory;
38   // Increment pheromone levels towards answerer
39   pheromones[node][category] ← pheromones[node][category] + y;
40   // Send feedback towards answerer
41   Send(f, node);
42 end

```

proportion of the total sum of pheromone values. As depicted in Figure 4.8, the probability of drawing a value in the range of a given link is directly proportional to the pheromone strength.

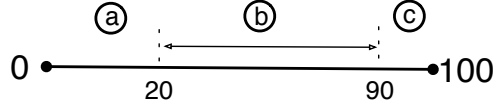


Figure 4.8: Probability of path selection.

Figure 4.9 presents a situation where the stigmergic protocol has 4 possible paths (*a...d*) for forwarding a particular question associated with category *cat*. In this example, node *a* represents a link which has previously generated an answer in this category via one of its own links and has also received positive user feedback as a result, earning this link the pheromone level of 0.91. Link *b* has also produced an answer to a question in this category via one of its connections, however, it did not receive any positive feedback and is therefore shown with a smaller star icon and the pheromone level of 0.11. Link *c* represents a path for which there has been no activities for this category and is therefore given the pheromone level of 0.06 – choosing such a link would be useful for exploration of the network and could uncover new experts. Finally, routing option *d* signifies a link which has been forwarded a question but has not yet generated an answer in response (as is the case in V3 of our protocol). This link has been reduced to the level of 0.01. Any of these four routes may be chosen, with a probability as given in Equation 4.2.

4.3.8 Routing Variations

All manner of subtly different or alternative approaches to the stigmergic tactics could be employed. Within the timescale of this thesis, many variations of the main algorithm have been considered, implemented and tested, yet only a handful proved to be useful and made it through to the final stage of evaluation. Initially, only a single category pheromone was used for each routing table entry, however this provided too much bias towards only the most popular of categories. In addition, initial approaches did not include feedback mechanisms, instead later choosing to provide feedback for only the most use-

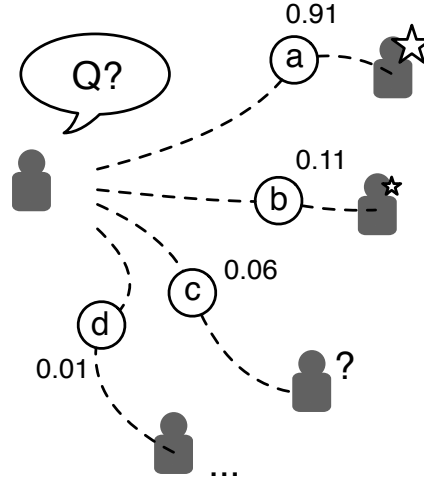


Figure 4.9: Stigmergic V1: path selection with scent levels.

ful answer and also for any useful answer.

The main variations considered in this thesis include:

- **V1:** a pheromone per category per routing table entry.
- **V2:** as V1 but with a local loopback routing table entry to allow self-learning of expertise at the protocol layer. Nodes will only be allowed to answer questions which are self promoted by the protocol.
- **V3:** as V2 but reducing routing pheromone strengths for a given link when selected for question routing, providing load balancing and additional network exploration or learning.

A loopback routing table entry (as present in V2 and V3) is helpful for overcoming the situation where experts are hidden behind less knowledgeable members of the network. All interested parties will attempt to answer questions, meaning they may be consumed before reaching a desirable user. The loopback allows for the underlying protocol to determine if a particular question is pushed up to the user for answering, rather than questions being automatically consumed.

Figure 4.10 shows a sample network as a question is generated from the central black node. Ideally a green node will answer, as they have matching

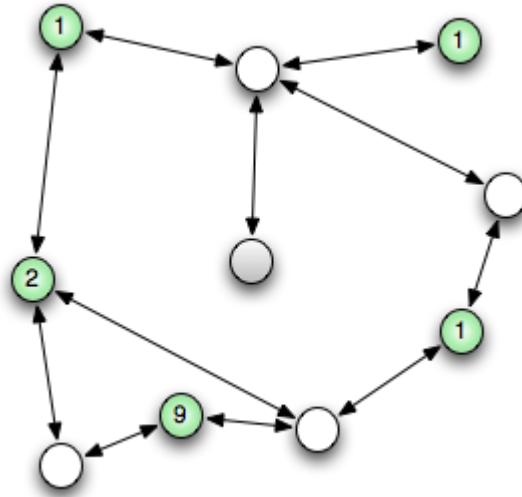


Figure 4.10: Skipping network users to reach experts.

interest categories to the newly generated question, with the expertise ratings denoted by numeric values. In the given diagram, in order to reach the better nodes, it may be necessary to travel through one of the more low-quality answering nodes. If our settings only allow for one or two answers, we may never reach the better nodes as the misinformed continuously consume the generated questions. Therefore, it is necessary to allow the algorithm to bypass poorly performing answering nodes over time.

V3 provides the additional characteristic of adaptation which prevents bombardment and several specific attacks. This will be discussed within the evaluation of this thesis.

When a question is generated at an arbitrary node, it is forwarded on through the network in the search for a suitable candidate for answering. In order to do this, it must travel through a series of nodes with varying profiles each of which define a set of interests and expertise ratings. The expertise ratings directly define the quality of all answers generated by the associated node.

The user model places any questions which pass through a node into a local priority queue for answering, only if the local profile states an interest in the question category. If a question has been stamped as requiring some number (>1) of answers, it will be forwarded on until this number reaches zero (with

the number decreasing each time it is placed in a local queue).

In order to reach the better active answerers in the network, the questions must somehow skip over the less knowledgeable network members. Those nodes with extremely low expertise values could be generating misinformed or spam answers which do not benefit the question asker and generally waste user time and effort, and network resources. This particular issue is tackled by V2 and V3 of the stigmergic algorithm where routing to self is considered alongside others, where the relative weight of pheromones determines the strength of the effect.

The routing algorithm allows pheromones to be negatively adjusted. Evaporation is included in the algorithm, where pheromone scent levels decrease in potency over time. Routing V3 also reduces the pheromone levels each time a question is routed through a link. The V3 promotes load balancing so that a particular link is not bombarded with requests and provides a means to penalise links which are yet to produce answers for outstanding questions.

To help improve the quality of the generated answers, *loopback* routing table entries (see Figure 4.11) have been investigated. When a question arrives at a node, it selects the next hop from its set of neighbours, including itself. Nodes can only attempt to answer questions which are *self-promoted* by the underlying protocol, and as such, the learning of an owning nodes expertise can take place automatically. Nodes will need to start with high pheromone values for those categories in which they deem themselves an expert to provide bootstrapping and initial bias. Over time, the local loopback pheromone entries may reduce, causing non-experts to be self-selected less often than more strongly scented external routes.

4.3.9 Learning and Warm-up Periods

When the Q&A network is first initialised before any interactions have taken place, the routing will perform in a similar manner to random routing. It is only after successful Q&A transactions have taken place that category pheromone values increase and can create preferential links towards specific areas of the network. Over time, the quantity of learned pathways will increase in proportion to the activities within the network. Warm-up periods are discussed in Section 6.2.2 of the evaluation.

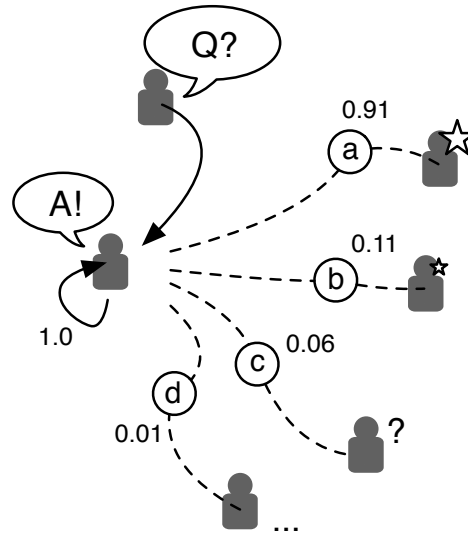


Figure 4.11: Stigmergic V2 & V3: loopback routing table entry.

4.3.10 Network Churn

Churn creates several issues for the routing approach. The networks of interest are random, therefore there is no inherent clustering of expertise within specific regions of the network. Learned routes are between two points in the network and thus breaks in the pathways cause routing which does not lead to the intended expert.

The routing should allow for alternative routes to be created to prevent single strong routes from emerging and to provide some robustness to the approach. Techniques such as *reducing pheromone levels on send* as seen in V3 should help to mitigate the effects of this particular churn-related issue by building multiple routes through the network.

It is also possible for routes on return pathways to be broken. When an answer is generated, it follows the route taken by the original question to the asker. If a node in the chain of links leaves then the connecting pathway will be lost. Return pathway will only break occasionally, but it is possible. In this work, when the return path is broken, the message is dropped.

4.3.11 Pheromone Evaporation

Typical stigmergic routing tactics apply evaporation, a means for reducing pheromone strengths over time. As pheromone strengths decrease, alternative routes emerge and previously good routes, which have deteriorated due to nodes leaving or congestion (in the form of a backlog or overloading at a particular node) are reduced.

Pheromone evaporation can be included in the algorithm simply by reducing pheromone strengths by small quantities at all nodes at set time intervals. Evaporation supplies negative feedback within the routing protocol, as does the routing variation seen in V3. Evaporation may be obsolete when V3 of the routing approach is used.

4.3.12 Pheromone Defaults, Maxima and Minima

Pheromones will need to start at equal levels in order to initially provide unbiased random routing. If routes are always to have some probability of being selected as the next hop, pheromones should never reduce to zero as this will eliminate a link from potential selection. Pheromones should not be allowed to increase above a sensible threshold, to ensure that a link does not consistently dominate (as seen in existing work preventing extreme differences in pheromone levels to dominate routing decisions [10]). We evaluate sensible values for pheromone defaults, maximums and minimums later in Chapter 6.

4.3.13 Oracle Nodes

Ideally no centralised systems would be required for the service to operate, realistically however, certain functionalities will need to be in a known and accessible place. Certain obstacles will need to be overcome such as the deniability of the service, exposing the entire network population and preventing a single point of failure³. An Oracle node or distributed set of Oracle nodes will be responsible for the following network support tasks:

Bootstrapping

A key functionality of a distributed service is being able to join and participate. The Oracle will need to provide a means of requesting addresses

³This is however outside of the scope of this thesis

of currently participating members to initially contact. This must be implemented to provide a uniform random network.

Neighbour requests

As the network exhibits churn, the Oracle will need to provide a means to request new neighbour addresses as existing connections are lost. Again, a uniform random network is required and therefore the selection must be random across all participants.

Maintenance Services

The Oracle may need to provide additional functionalities such as distributing up-to-date category ontologies. Such service updates could be included in the bootstrapping process or as part of a scheduled periodic update across the network.

For this work, an Oracle will be included to provide these key functionalities without investigating solutions for the issues and problems presented here. The Oracle will allow any node in the network to request new neighbours, bootstrap into the network and to be used for any additionally required functionalities.

The oracle system presents a clear threat to the privacy of the network, however it is used in this case to aid simulation. In a real world scenario it would be possible to create question and answer networks without the need for such a system.

4.4 Attack Models

The ad hoc Q&A networks may be attacked and abused in various ways. New protocols and routing tactics have been designed to thwart these potentially malicious attacks.

4.4.1 Establishing Author Identity

Question asking and answering needs to be plausibly deniable, and as such, the identity of an authoring node is disguised via an unknown intermediate pathway. The random network topology and underlying protocol hides this information and so inherently, it is protected. The Time To Live (TTL) values of messages are drawn from a random distribution, meaning that the source cannot be identified by its immediate neighbours. The content of the questions

and answers will not be controlled directly, however, an anonymizing layer could be utilised to remove traces of identity such as names and addresses.

4.4.2 Reducing Answer Quality

Users may try to reduce the quality of answers generated by the network by providing positive feedback for bad answers, thus reinforcing paths towards non-expert users. This would not exclude other users however, due to probabilistic nature of the routing, the effect would subsequently be diluted by other interactions. Feedback is tied to question IDs, and hence a question and one of its answers, so that spurious feedback cannot be generated.

Another possibility is that intermediates in a question and answer exchange generate false feedback. By this a malicious user could honour an exchange, but decide to generate positive feedback for any or all answers received. This form of attack would be similar to always submitting false positive feedback, however, it would only manipulate a portion of a pathway between an asker and answerer.

4.4.3 Eager Answerer

An eager answerer could answer all questions directed at them in an attempt to cheat the routing system. If a user decides to answer all the questions which arrive, they may be able to negate the advantage of positive feedback sent to neighbouring nodes. This form of attack could lead to a node increasing the likelihood of being sent questions without having any expertise in the subject matter, just the time and attention required to deal with the incoming queries. In the standard approach, a malicious user who is generating junk answers counteracts the pheromone adjustments caused by positive feedback by supplying enough answers (ω) to satisfy this condition, namely:

$$\omega = (PHEROMONE_FEEDBACK/PHEROMONE_UPDATE) + 1.0$$

The ω value is used to define the number of answers (regardless of quality) that a user needs to submit to exceed the pheromone increase caused by positive user feedback.

For example, if the pheromone feedback rate is 0.80 and the update rate is 0.05 then a malicious user would need to answer over 16 questions to create the attack. This is indeed an important attack on the routing mechanics. Fortunately, using V3 of our approach we are able to set a suitable strength

decrease when sending questions to negate this form of malicious attack, such that either a much higher number of answers is required or, making this impossible by penalising pathways producing solely junk answers. For example, reducing the strength to the links which questions are forwarded through by an equal or greater amount than the enforcement gained from routing back an answer.

4.4.4 Denial of Service (DoS)

Malicious users could bombard the network users by flooding the network with questions or answers. This form of attack pollutes the network with traffic aimed at consuming user time and attention. This attack can be controlled by rate limiting the number of forwarding requests honoured on behalf of a specific neighbour. Nodes which exceed some threshold can be ignored or even removed as a neighbour. Nodes supplying spam as questions will eventually be routed out via our routing approach seen in V3, which allows link strengths to be reduced when using links which are not generating useful answers. This threshold or rate does not need to be determined prior to the network construction and could be enhanced through end application features such as signals from the end users.

4.4.5 Colluders

In an attempt to cheat the network mechanics, colluding users may try to reinforce routes between one another. They may ask questions regarding specific subjects and when answered by a fellow colluder, provide positive feedback. This allows a colluding set of nodes to control question flow towards specific users, undermining the routing. However, random routing and probabilistic link selection makes it hard for this attack to have a widespread impact on the network. The load-balancing mechanisms of the V3 model further reduce the effect of collusion.

If the Oracle system is malicious it is able to provide the most powerful attack, as it may provide the bootstrapping node and neighbours for a given target node. The Oracle may allow a form of attack in which an unsuspecting user is surrounded by attackers. A distributed Oracle system would be beneficial to restrict this form of attack.

4.4.6 Author Identification

When using such a system over prolonged periods of time, the author of a specific question or answer could be identified by means of authorship attribution (AA) [88, 89]. With such techniques, the questions and answers for a given author could be linked together by analysis of the words used, the order of appearance and writing styles. Although such tactics would not directly divulge the identity of a particular author, it could build a set of evidence to aid author identification. As author plausible deniability is the key requirement, this form of attack will not pinpoint an exact individual.

4.4.7 Encryption

It is assumed that the textual content of questions and answers is plain text however, as described above, this could create additional means to help identify an author and indeed may give away more information to intermediates than necessary. One such additional method would be to employ the use of asymmetric encryption. On the creation of a new question an author could encrypt the content body using a one-time public private key pair. Instead of directly announcing the question, the author could choose to use a **REQUEST** protocol message to locate an answerer which includes the one-time public key. If a particular answerer wishes to provide an answer they could submit an **ACCEPT** protocol message which includes their own unique one-time public key. On receipt of an **ACCEPT** message a user could encrypt the question with the public encryption of the answering node, and proceed normally. On receipt of the actual question the answerer would encrypt their answer using the original public key of the originating node.

Ultimately such techniques would increase the level of resources and computation required across the board but may supply additional privacy.

4.5 Summary

This chapter has presented the stigmergic routing technique and the related protocol and attack considerations.

The stigmergic approach towards question routing in Q&A networks provides deniability, as the next hop is determined at each node in the network without the use of identities. The stigmergic approach reinforces routes which

produce good answers and increases the likelihood of the path being re-selected in the future for those questions which happen upon some section of a previously proven route.

It is now possible to consider the simulation of realistic Q&A networks using the ideas and techniques found in the preceding chapters. Evaluation and comparisons can then be made via a series of suitable experiments.

Design & Simulation

This chapter presents the systems, entities and procedures required to evaluate the properties of a distributed Q&A service over peer-to-peer networks. Using the user model and stigmergic routing details from the previous chapters, as well as exploring new techniques and principles, a working platform for the simulation of distributed Q&A networks will be described. Suitable performance metrics are presented to allow evaluation of various routing approaches. The described Q&A platform implements a fully decentralised networking environment, without branding questions and answers with the authoring node's identity, in order to support meaningful evaluation of the proposed routing techniques.

5.1 System Design

A distributed Q&A network requires a range of entities, components and features, which are described below.

5.1.1 Nodes (Physical Devices)

Nodes (represented in Figure 5.1) within the Q&A network represent the devices on which the service will operate, for example, a mobile device such as a mobile phone, or desktop, laptop or notebook computer. Devices will have varying degrees of mobility which presents a challenge however, that is not the focus of this thesis. Nodes will connect to other devices to form overlay networks via the Internet. The specific networking device below IP should not matter. Nodes will be capable of running the Q&A service, regardless of the specific device type.

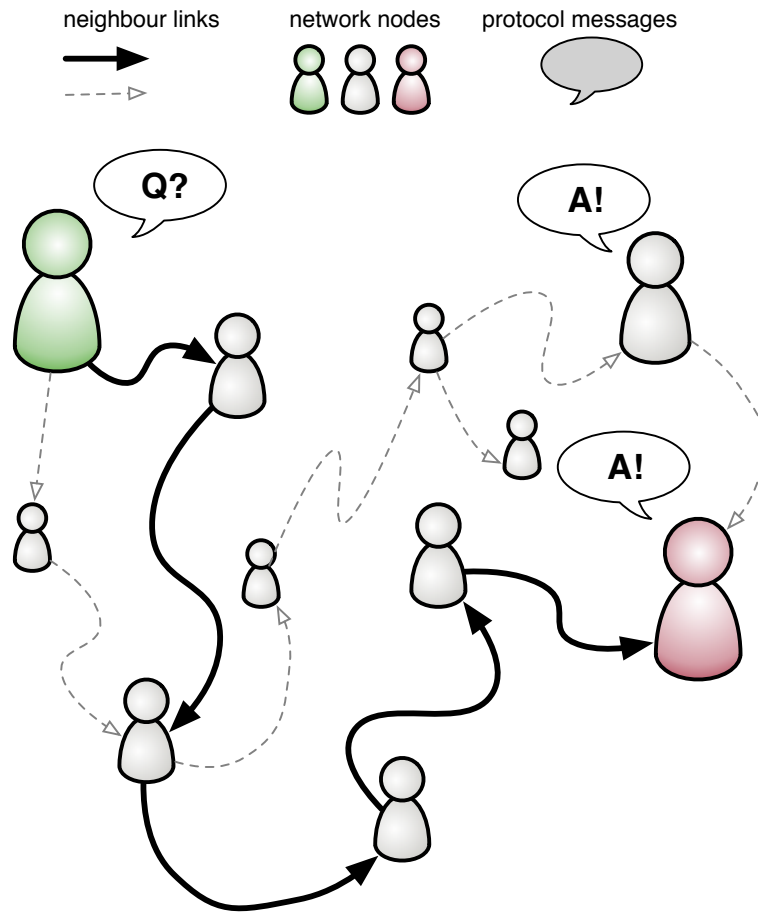


Figure 5.1: An example of a small Q&A network.

5.1.2 Users

Each node will have exactly one user, and each user will have only one node which they occupy and operate. Users are the human element of the system and make up the Q&A service content. Figure 5.1 shows network nodes as user icons to represent the relationship between both node and user.

As previously explored, users within the system will have a range of interests and expertise (categories in which they ask questions and give answers) and these may overlap. For example, a user may be interested in both *Computers* and *Gardening*, but only have expertise in *Gardening*, or indeed in a different category altogether. Users follow a Markovian model of attention,

allowing them to transition between states of inactivity and attention during the duration of a session.

Users will have the ability to generate questions and answers, and will receive expertise ratings within the context of Q&A. These values will be drawn from the range of possible best answer counts found within the dataset per expertise category.

5.1.2.1 Local Priority Queue

Each node has a local priority queue that holds questions awaiting an answer, ordered by the interest level by the given user. Questions within the local queue are answered in turn and are of a fixed size. When a question is added to a full queue, the least interesting question (the last item in the queue) is dropped and discarded if the new question has a greater perceived level of interest from the user.

5.1.3 Protocol Messages

A simple collection of protocol messages are used within the Q&A network in order to support the core functionalities. The protocol messages are the objects that are transported around the network between nodes in order to serve the network users' Q&A requirements. Protocol messages are represented in Figure 5.1 within speech bubbles.

5.1.3.1 Questions

Questions exist as snippets of text that are generated by authoring users and sent into the network via an author's node. A question is stamped with a specific question category, and has an associated length in words. This is used to indicate how long it took to compose and the time it takes the question to be read by other users in the network. The message structure of a question can be seen in Table 5.1.

5.1.3.2 Answers

Answers, like questions, are user generated text snippets which relate to a specific question and also have a unique GUID. An answer includes the original GUID of the related question so that it may flow back towards the question asker, but also has a unique GUID of its own to allow for a path between the

Field	Data Type	Length	Comment
GUID	java.util.UUID	–	global identifier
TTL	int	32-bit	decremented per hop
Qtext	java.lang.String	–	question
length	int	32-bit	length in words
cat	int	32-bit	category

Table 5.1: Question message structure.

question asker and the answer to be created. Similarly, an answer has its own length in words. The *cat* field is used to represent the type of category that the original question related to, allowing for the correct category pheromone increases to be made. The message structure of an answer can be seen in Table 5.2.

Field	Data Type	Length	Comment
GUID	java.util.UUID	–	global identifier
QGUID	int	32-bit	original question
Atext	java.lang.String	–	answer
length	int	32-bit	length in words
cat	int	32-bit	category

Table 5.2: Answer message structure.

5.1.3.3 Feedback

This message acts as a means for users to provide positive feedback in response to a given answer. For the Q&A stigmergic elements to work, user feedback is supported in a unique system protocol message. A feedback message includes only the GUID of a specific answer and works in a similar way to backwards ants as seen in previous stigmergic works by Barán [87], supplying the feedback elements found in stigmergic routing algorithms. The feedback messages will follow the route taken by their associated answers, back towards the answer generator. The message structure of a feedback message can be seen in Table 5.3.

Field	Data Type	Length	Comment
GUID	java.util.UUID	–	global identifier
AGUID	int	32-bit	related answer
cat	int	32-bit	category

Table 5.3: Feedback message structure.

5.1.3.4 Joins and Neighbour Management

For completeness, the protocol includes messages for joining and requesting new neighbours from the network. Where possible, a join request is honoured by a node to provide a means for new nodes to enter the network. Neighbour requests can be probabilistically accepted or rejected by nodes, depending on their current volume of connectivity.

In this work, joins and neighbour protocol messages are removed. We want reproducible control over the random network topology and to reduce the associated overhead which would have been irrelevant to the core of this study.

5.2 Performance Metrics

To evaluate the routing approach within Q&A networks, we consider a set of metrics over different aspects of the solution quality.

5.2.1 Answer Quality

It is important to identify how good an answer is in comparison to all other possible answers that *could* have been generated by the current network of users. Assuming that answer quality depends directly on user expertise, the following description and precise definition of quality can be used:

When a question is first sent into the network, those active users who have an interest in the category (set A) are recorded. When an answer is generated, these nodes are checked to see if they are still online (set B). The set $C = A \cap B$ is then found, consisting of those nodes who had the potential to answer a particular question. The available expertise is then the set of user expertise ratings from C . Set C contains the distinct set of expertise ratings, allowing for one or more nodes to be represented by a single element. For example $C = \{0, 3, 4, 35, 100\}$. A ranking $0 \leq \text{quality}_a \leq 1$ is then produced based on the members of C to tag an answer with a *perceived answer quality rating*

via Equation 5.1 where a refers to the answer in question and u the authoring node of a . If an answer is authored by the expert with rating 35 in the above set the quality rating would be $r = 0.75$. We use the average and best answer qualities for a given question to evaluate the quality of answers generated by the network.

$$quality_a = ranks_a * \frac{1}{|ranks| - 1} \quad (5.1)$$

We note that a low TTL can result in unreachable nodes forming part of set C. The practical trade off in TTL choice is analysed in the results (see Section 6). Furthermore, if you factor in churn, broken routes which exist between the asker and answerer may emerge.

5.2.2 Attention Consumed

Attention consumed refers to the quantity of time each user spends reading questions, composing answers and dealing with received answers. An ideal algorithm will consume a low amount of attention for each question, and a consistent amount from each user. An algorithm that maximises answer quality may focus all questions on a few users. In order to achieve a balanced attention cost, some reduction in quality may be required. The attention consumed by the network as a whole, per question and per user, is used to present an overall picture of the user-related costs.

5.2.3 Percentage of Unanswered Questions

For each question in the Q&A network, all received answers are linked back to the original question via its GUID. All questions which have zero answers before a user leaves the network are deemed *unanswered*. This is an important metric in determining the usefulness of the approach, as the usefulness of a Q&A network will rely in part on the ability for questions to be answered at all. Unanswered questions do not contribute to our answer quality metrics as we are only interested in the quality of answers being generated.

5.2.4 Path Lengths

Each question tracks how many hops it has taken throughout its lifetime. When an answer is generated at a node, the number of question hops is

recorded and logged against it. This allows for the analysis of the number of hops for each question and answer pair.

As well as reducing network load, the aim is to reduce path length to avoid broken pathways. A broken pathway can be defined as when the original pathway between question asker and answerer contain a missing hop, as churn causes the network topology to change over time. These broken pathways cause answers to be lost and dropped, meaning wasted effort and lost answers¹.

5.3 Simulation

This section provides details regarding the software and configuration of our simulator. The simulation element of this study was a learning experience and as such generated many obstacles and problems that had to be overcome along the way.

5.3.1 PlanetSim

The PlanetSim simulation framework [90] was chosen as the test bed for our experiments. PlanetSim provides a discrete event simulator via an extensive API and examples developed in Java. The main motivation behind this decision was that the most established and well known simulation platform, Network Simulation (ns-2)², included too many complexities in regards to low level networking representations. The ns-2 examples, support and tools are extensive, however, the target networks of interest are large and the underlying network mechanic are not the main focus of this study. Several other P2P simulators (including PeerSim [91], P2PSim [92], OverLay Weaver [93] and OverSim [94]) could have been used, but PlanetSim was a suitable option [73] at the time. During the course of this study I became a developer for the PlanetSim team³.

5.3.1.1 Layered Approach

The PlanetSim framework is divided into several layers including: **Network**, **Overlay** and **Application**. The **Network** layer allows for the creation and

¹Techniques and approaches to negate the effects of broken pathways has been left for future work and is outside the scope of this thesis.

²<http://www.isi.edu/nsnam/ns/>

³<http://projects-deim.urv.cat/trac/planetsim/wiki>

management of a network, including the addition and removal of nodes, simulating the processing of messages across all network nodes and gaining access to important metrics such as the current network size. The **Overlay** network consists of the nodes within the network, which will be discussed in the next section. The **Application** layer allows for various application functionalities to be installed and included on network nodes. The **Application** layer was not required in this work and was merged into the functionalities of the network entities themselves, which all operate in an identical manner.

5.3.1.2 Node and Node Handles

PlanetSim provides abstractions to allow the representation of a **Node** in a network and are addressed via a **NodeHandle**. A **Node** can communicate via **Messages** if the corresponding **NodeHandle** is known. From this simple set up, a developer may extend a **Node** to create desired functionalities in order to represent any form of distributed system.

5.3.1.3 Message Queues

Each **Node** has an incoming and outgoing queue ordered in a first-come-first-served (FIFO) style. Nodes process the items within incoming queues during simulations, and the resulting messages are placed in the outgoing queue. This element of the simulation framework can be considered as a general abstraction of the process networked devices use to communicate. A user with a node and incoming (green) and outgoing (red) queues can be seen in Figure 5.2, where the user may ask questions and consider and compose answers, which are sent into the Q&A network via the device represented by a node. In addition, the users separate priority queue (blue) is also presented in Figure 5.2, which houses the questions awaiting an answer.

5.3.1.4 Simulation Steps and Lengths

Time is simulated in discrete steps, whereby each node can process some number of messages at each step from its incoming message queue and generate new outputs in outgoing queues. A simulation will last a number of steps and at each step every **Node** in the network is processed. A simulation step can represent a unit of time and in this work it is considered to be a single second of real time.

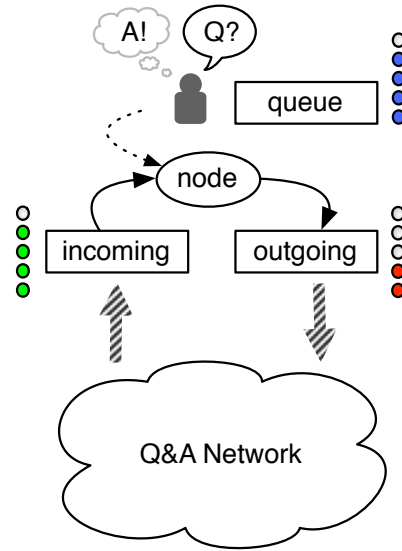


Figure 5.2: User with node and queues.

With a network in place, time can be simulated in single or multiple steps as and when requested. During a simulation step, each node is processed in turn triggering the modelled users who generate the events and messages.

The processes which take place at each step can be abstracted to those seen in Listings 5.1 and 5.2, where each node in the network is called via an Iterator and the process method is called to manage the various requirements such as user attention transitions and question composition progress.

Listing 5.1: Simulation Step Process

```

while(currentStep<stepsToRun) {
    doChurn();
    processAllNodes(network);
    // Runs the process to simulate one time step
    network.simulate();
    currentStep++;
}

```

5.3.1.5 Route Messages

Network messages are packaged within a **RouteMessage**. These messages act as a wrapper for protocol messages and it is this abstraction that is used for message passing between nodes. When a message (for example a question or answer) arrives at a node it is contained within an outer **RouteMessage**. These wrapper messages are reused by PlanetSim to improve performance. Failure to realise this feature of reusable wrappers can cause an array of issues such as unexpected message reuse (the housed object changing over time). The wrapped objects contained within the **RouteMessages** always need to be removed before passing to node methods and functions. A selection of the key messages used appear in Figure 5.4.

Listing 5.2: Processing all the Network Nodes

```
processAllNodes(Network n) {
    Iterator it = n.iterator();
    while(it.hasNext()) {
        Node n = (Node)it.hasNext();
        n.getProfile().process();
    }
}
```

5.3.1.6 Behaviours

A very desirable feature of PlanetSim is the ability to define **Behaviours**. This allows for the definition of a particular function to handle a particular protocol message, and for fairly rapid prototyping at the stage of adding new protocol messages. The **Behaviours** are defined by creating a new subclass of the **Message** Java class and handling code, both of which are later referenced in the main configuration file for the simulation.

5.3.2 Networking Topology

A new overlay network can be created via the **Factories** supplied by PlanetSim. An overlay network consisting of a random topology where x nodes can be requested and the resulting **Network** is realised. Topologies can also be created manually, by triggering a new node to join via any existing bootstrap node in the network. To have a more fine grained control over the network

topology and the exact random seeding, the manual method of generating a network topology was adopted.

5.3.3 Node Structure

The network nodes are composed of a number of key elements that are represented in the simulation. A high level abstracted view of the elements that make up a single node can be seen in Figure 5.3⁴.

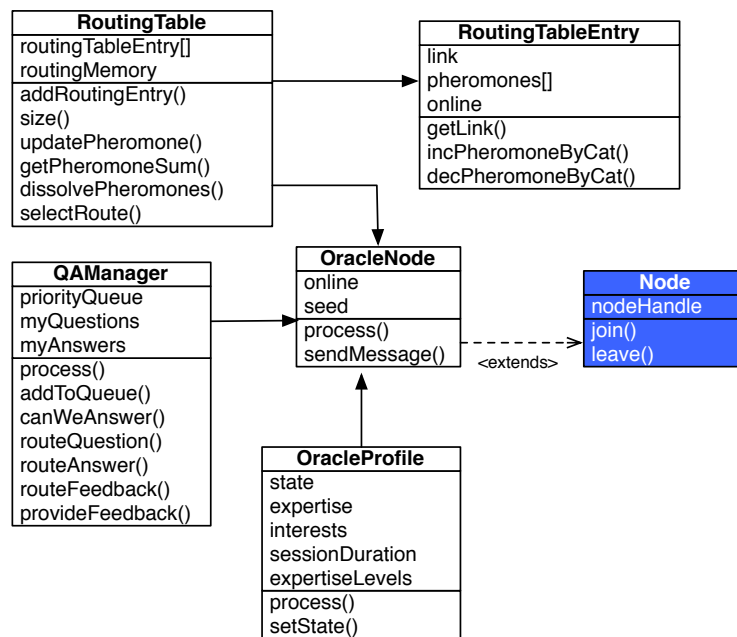


Figure 5.3: High level abstracted node structure.

The **Profile** element of the node is responsible for holding details relating to the simulated user. This includes the Markovian state model discussed in the previous chapter, so that the user transitions between states of attention and inactivity. In addition, details of the categories the node has an interest and expertise in, their expertise levels per category and details regarding consumed levels of attention are stored.

QAManager is used to manage the routing of questions, answers and feedback through the network. The previously mentioned **Behaviour** classes of PlanetSim will call methods from the **QAManager** section of the node. Routing

⁴Blue nodes represent the elements of PlanetSim.

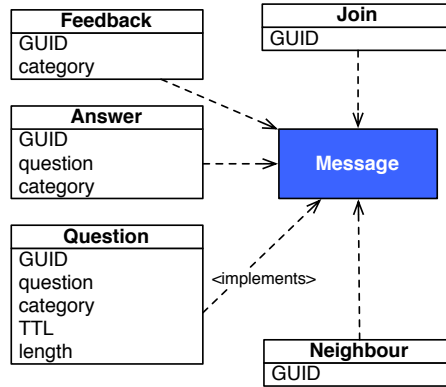


Figure 5.4: High level abstracted message structure.

decisions and histories are recorded in a `RoutingMemory` which resides in the *RoutingTable*.

Known neighbours are recorded in the `RoutingTable` in the form of `RoutingTableEntries`, which compose both an address and related Pheromone Category levels.

5.3.4 Question Generation

Users generate questions probabilistically based on some constant probability. The successful event will cause the node to spend some period of time (simulation steps) composing a question. When completed, the question will be sent into the network via a neighbour chosen by the routing algorithm in use.

5.3.5 Answer Generation

When a question is routed to a node and added to the local priority queue it is dealt with as and when the user has time. Questions are ordered by priority so that the most interesting is answered first. An answer takes some time to compose and this can only take place while the user is paying attention within the network (Markovian model state). Once the answer has been composed it is generated and submitted back into the network along the path it arrived.

5.3.6 Feedback Generation

A user will generate positive feedback for answers that have arrived and been read successfully (the reading of answers takes some time as a function of the answer length and reading abilities in WPM). When feedback is required, a feedback message will be generated and sent back into the network towards the source of the related answer message.

5.3.7 User States

Users will follow a Markovian model, as discussed in the previous chapter, to transition between two possible states: *paying attention* and *idle*. While in the idle state, users will not generate or read questions, generate answers or provide feedback. While paying attention a user may perform all of these tasks, with question asking taking priority. It is during this attention state that user attention is consumed.

5.3.8 Network Churn

As discussed in the previous User Modelling chapter, user session durations (the amount of time they stay) and inter-arrival times (interval between arriving nodes) can be modelled using the Weibull Distribution. The Apache Commons Mathematics Library⁵ contains numerous distributions that can be used for this purpose. The Apache library is simple, efficient, tested by the community and in active use.

Nodes are assigned a session duration when they enter the network based on a specific random number drawn from a Weibull distribution using the Apache library. Nodes arrive in intervals separated by an inter-arrival time which is also determined by a random number from a Weibull distribution.

Using the Apache library, a Weibull distribution object can be created and used to generate a value as seen in Listing 5.3. This library is extremely useful as the process is very straightforward to implement and use.

With a given Weibull distribution producing session durations and inter-arrival value in minutes, we can simply convert to seconds for use within the simulation by multiplying the values by sixty (converting to seconds/steps from minutes).

⁵<http://commons.apache.org/math/>

As depicted in Figure 5.5, a node a arrives at some point in time and is assigned a session duration. At a later point in time node b arrives, followed by the departure of node a when its session duration expires.

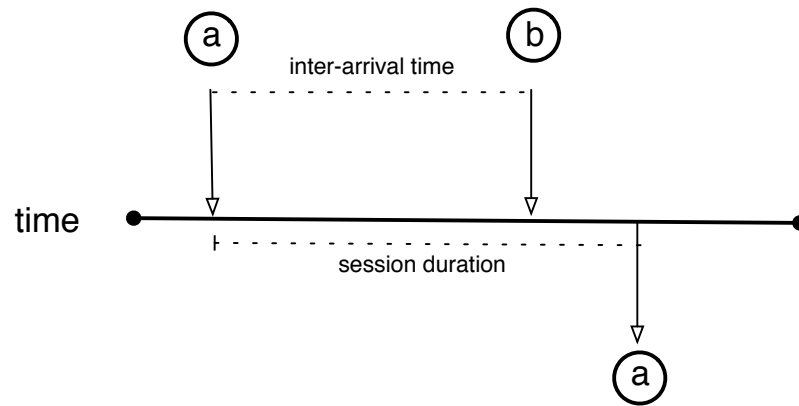


Figure 5.5: Churn: Inter-arrival time and session duration.

Listing 5.3: Weibull Usage in Java

```

java.security.SecureRandom sr = new SecureRandom();
double d = sr.nextDouble();
w = new WeibullDistributionImpl(k, λ);
double result = w.inverseCumulativeProbability(d);

```

5.3.9 Random Number Generators and Seeding

Random Number generators are used extensively in this study to draw values from discrete and continuous distributions. In order for the probabilistic modelling to be accurate, the random numbers used must exhibit a uniform distribution. In the past, the authenticity of the sequence of numbers generated has caused great concern, however, Java supplies some useful methods for generating uniform random numbers, such as `java.util.SecureRandom`. This generator will return *“the next pseudorandom, uniformly distributed double value between 0.0 and 1.0 from this random number generator’s sequence.”*⁶. This can be confirmed by plotting the values returned for some large number of calls (see Figure 5.6) where the values are placed in 0.01 range bins over 50,000 unique calls using the `nextDouble()` method.

The seed of a random number generator determines the sequence of numbers which are generated from it. Various seeds need to be evaluated for experimentation, but testing various approaches under the same conditions requires repeatability. Fortunately, a seed can be specified via the `java.util.SecureRandom` class via the example in Listing 5.4. In this example the Secure Hash Algorithm (SHA-1)⁷ is used with the seed *sd* to prime the `SecureRandom` object *sr*.

Master seeding must be used to correctly seed all other seeds used in the simulations, for example seeding each users question asking generator and probabilistic routing techniques. This master seeding will allow repeatability as and when required by priming each generator in a set order from the master seed.

⁶[http://download.oracle.com/javase/1.4.2/docs/api/java/util/Random.html#nextDouble\(\)](http://download.oracle.com/javase/1.4.2/docs/api/java/util/Random.html#nextDouble())

⁷<http://download.oracle.com/javase/1.4.2/docs/api/java/security/SecureRandom.html>

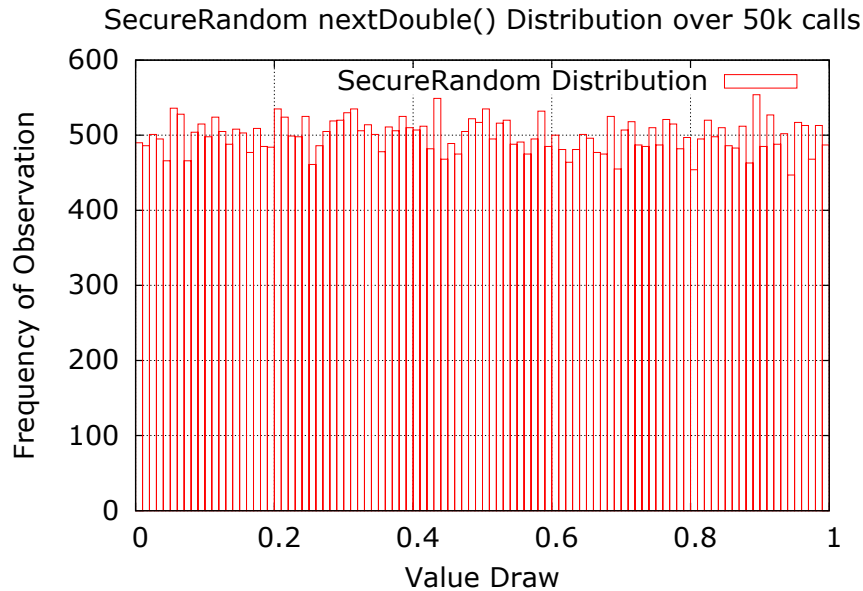


Figure 5.6: java.util.SecureRandom value distribution.

Listing 5.4: Weibull Usage in Java

```

try {
    sr=SecureRandom.getInstance('SHA1PRNG');
    sr.setSeed(sd);
} catch (Exception e) {
    System.out.println('failed to seed.');
```

5.4 Experimentation

In order to generate results and statistics concerning experiments of the simulated Q&A networks, a suitable collection of processes and procedures are required to set up and run a scenario.

5.4.1 Data Distributions

Continuous distributions are drawn from the Apache Commons Mathematics Library, while discrete distributions are represented in a static continuous array form. The normalised distributions are used to create cumulative distributions

from which values may be drawn via the use of a uniform random variable.

5.4.2 Configuration

Due to the workings of PlanetSim, a configuration file was created to house the variables which are used for the simulations (A example configuration file can be seen in Appendix 8.1). These name-value pairs are loaded at run time before the simulations begin and assign values to the key system variables. This tactic allows for the adjustment of simulation variables from a configuration file, rather than them being hard-coded in a static manner. Key variables of interest that are loaded from the configuration file include:

- QRATE: Question asking probability, per node, per step.
- REQ_ANS: Number of answers required per question.
- POP_SIZE: Population size.
- INIT_SIZE: Initial network size.
- WEI_LAMBDA&K: Weibull distribution parameters.
- SIM_LEN: Simulation length in steps.
- ITERATIONS: Number of iterations.
- SEED: Master seed value.
- P_VALS.*: Pheromone update parameters.
- QUEUE_SIZE: Local queue size.

5.4.3 Network Set up and Creation

An experiment program is used to manage the associated aspects of running a simulation, which allows for different configurations and experimental practices to be adopted, for example, the examination of a single variable under various ranges.

A population of possible users with associated traits is generated in the form of a user profile. From the population of users, a set number are randomly selected and used to create an initial network (see Figure 5.7). Each node will

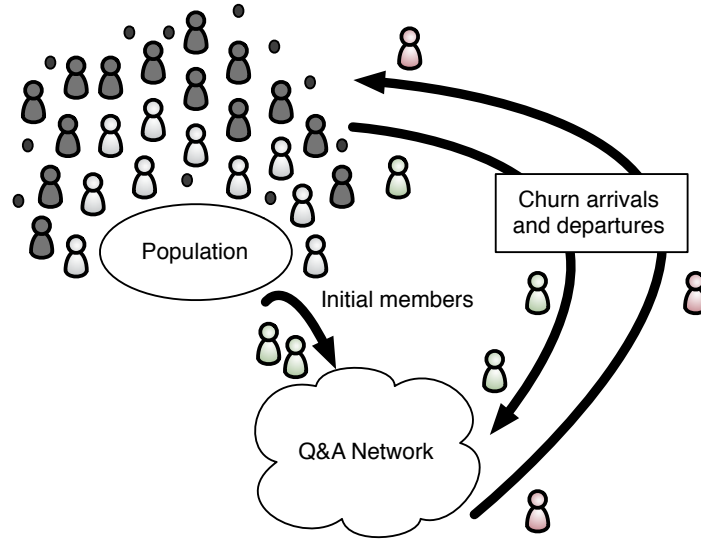


Figure 5.7: Initial network members.

have a set of neighbours uniformly assigned from the entire network, creating a random network topology. Once created, the simulation is ready to begin.

A simulation runs for the number of steps specified in the configuration file. A simulation is typically set up to represent several hours of time and takes time to complete depending on the number of nodes, question asking probability and desired simulation length.

Simulations will generate a set of seeds for the number of iterations required. Each question routing approach will be attempted in turn for the number of iterations dictated for the defined simulation length. When each approach has been completed, the simulation will deal with the generation of results and then cease.

As simulations can take many hours to complete, they are left to run independently on dedicated computer facilities, often overnight.

5.4.4 Gathering Results

In order to correctly draw analysis of the various approaches for question routing, results must be collected during the course of a simulation for the measures discussed in Section 5.2. To provide a more accurate picture of network performance, approaches will be compared across a number of iterations, the

combined results across several uniquely seeded simulation runs. This multiple iteration approach provides greater statistical strength to the results. Each iteration for each approach will be under identical question asking rates, user states, network set up and neighbours.

When a node generates a question it is passed to a results encapsulating module. Each answer that is generated in response is again passed to this module, associating it with the relevant question. This allows for statistics regarding answer quality, number of answers, unanswered questions and path lengths to be collected for evaluation.

Statistics relating to percentiles are of particular interest here as the networked conditions are slightly different under each unique iteration and include multiple runs.

At the end of a simulation run, question asking is disallowed and the network is allowed to continue running until no messages are in queues or in transit and users are not producing answers. In other words, simulations continue until their natural conclusion.

5.4.5 Processing Results

Statistics are collected during the simulation runs of each approach and iteration, and at the end of the simulation the results are processed. A unique results directory is created for the run, which records the configuration parameters used for the experiment including the seed used.

A statistical analysis of the data is then created, for example, by generating the 5th and 75th percentiles of answer quality. These values are recorded to a data file for each approach and an associated gnuplot⁸ script is created referencing the file. Finally, several unix bash script files are created to allow the processing and generation of graphs and results as and when required. An example gnuplot script and bash script file can be seen in Appendix 8.2 and 8.3.

The raw results of the experiment, along with the processed graphs, are available at the end of the simulation and can be copied and accessed via the University of Sussex webspace if required.

⁸<http://www.gnuplot.info/>

5.4.6 Experimentation Debugging

The simulated Q&A network scenarios took a long time to design, develop and debug. Numerous issues and obstacles had to be overcome. One major issue was network size, as the larger the network, the more difficult it becomes to trace and consider the pheromone values and small logic errors. Another complexity is absorbing and understanding the pheromone state information from text outputs alone. These two problems together, coupled with the long simulation times and user models, make the simulation process a time consuming and fairly troublesome affair.

5.4.6.1 Step Through

To ensure that the master seeding approach creates identical networks and user behaviours, a low level evaluation of key system outputs such as network size was validated and verified by hand. Although time consuming, this was extremely valuable. For verification, churn levels were inspected by comparing the network size over time to a stand-alone application using an identical seed.

5.4.6.2 Network Visualisation

Being able to visualise the Q&A networks has been invaluable. Having the ability to see the nodes and connections in visual diagrams significantly aids the understanding and formation of solutions. When custom routing tables are added to the equation, appropriate visualisation tools become increasingly useful to understand emergent overlays.

GEPHI⁹ was used to visualise hand coded .gdf output files. This allows for weighted edges between nodes and makes layout and statistical information available. Figure 5.8 shows a 35 node network at the end of a simulation run with edge weights decided by pheromone strengths for a specific question category (numbers represent associated expertise levels in the category with fullstops representing a node who is not interested in this category). Figure 5.9 shows a larger network of 100 nodes.

⁹<http://gephi.org/>

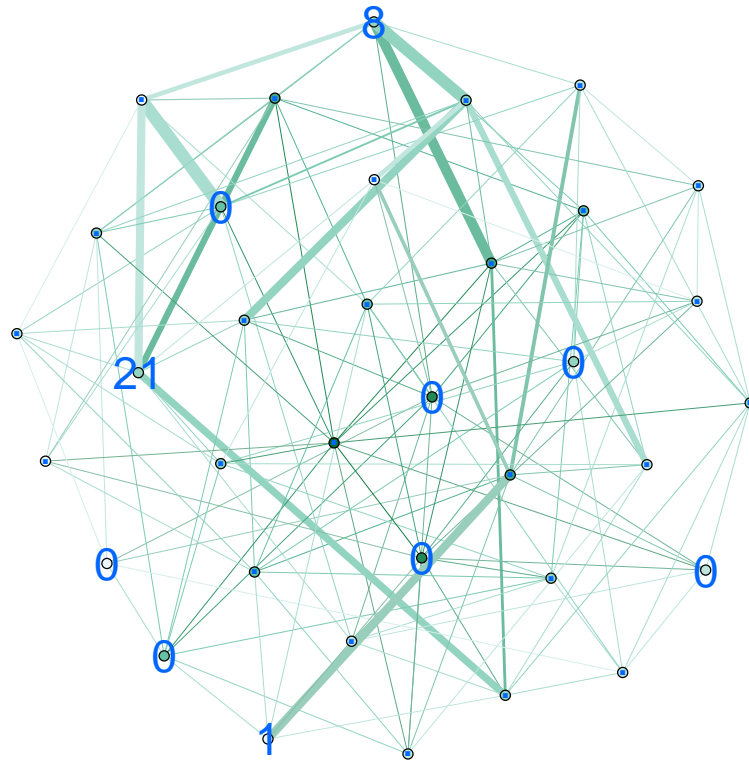


Figure 5.8: 35 Node visual with pheromone scent levels.

5.5 Summary

This chapter has presented details relating to the implementation and simulation of ad hoc Q&A networks. Details regarding a simulation framework and its operations and considerations have been included. This work is essential to outline several naïve approaches to the task at hand and to reinforce the constraints and requirements for this study. The procedures and processes

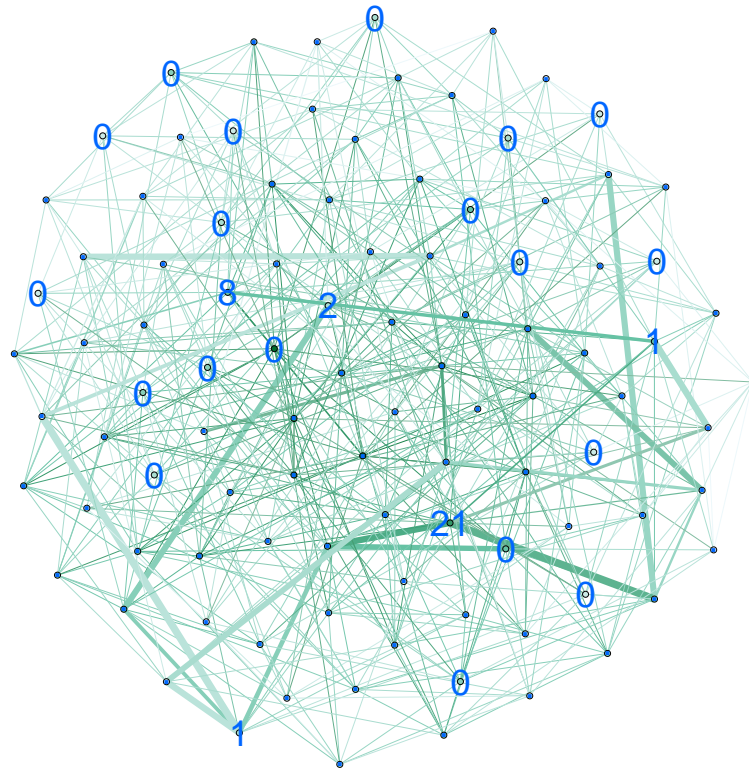


Figure 5.9: 100 Node visual with pheromone scent levels.

required for simulation and evaluation of the question routing approach have been presented to allow for an in-depth study to take place.

With the Q&A simulation and experimental set up in place, it is now possible to evaluate and compare the routing approaches in the next chapter.

Evaluation

This chapter explores and evaluates the various routing techniques and scenarios. The stigmergic-inspired approach is compared with the random and flooding methods where appropriate. To reiterate, random benefits from having a low overhead and even distribution of user attention. A flooding approach demands the maximum levels of user attention and effort from the network, while locating both the best and worst answerers.

The evaluation begins by exploring simulations of specific network sizes and simulation lengths, followed by pheromones and generic protocol requirements, the impact of churn and finally, results for a realistic combination of parameters. The aim of this Chapter is to understand how the variables affect the key metrics of interest in the Q&A networks and demonstrate the benefits of the stigmergic routing approach.

6.1 Network Size and Simulation Length

To confirm the challenges with simulating large networks for long periods of time, this section analyses the time and memory requirements. In order to compare and contrast the approaches it is important to choose simulation lengths and network sizes which complete in a realistic and reasonable time period. Throughout the course of this thesis the simulations have often taken several days to complete, due to the various approaches, network sizes, question asking rates, multiple runs for statistical strength, churn scenarios and iterative design.

To perform this comparison the same shared linux-based computer was used. This machine is a forty-eight core (Intel®Xeon®CPU) machine with 252 gigabytes of memory. It is part of a cluster which may be running multiple

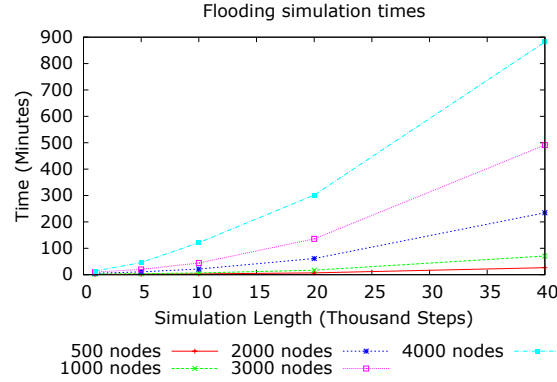


Figure 6.1: Flooding simulation times.

jobs.

Each simulation uses an identical set of seeds such that the network structures, question asking rates and user models were identical. These results include the system resources used to record the results and all other associated simulation essentials. Results are presented as arithmetic means over 5 iterations.

The flooding, random and stigmergic approaches are compared for network sizes ranging from {500, 1000, 2000, 3000, 4000} for simulations of {1000, 5000, 10000, 20000, 40000} time steps (up to approximately 10 hours of simulated time) in length.

These results make use of the final simulation parameters (e.g. question asking rate, required answers per question, pheromone rates) presented later in this chapter.

6.1.1 Running Times

As shown in Figures 6.1 through 6.3, the simulation time requirements rise as a function of network size and simulation length. The time requirements for larger networks are significant for the flooding approach due to the larger proportion of messages and subsequent user activity. The flooding approach is the real bottleneck when drawing comparisons.

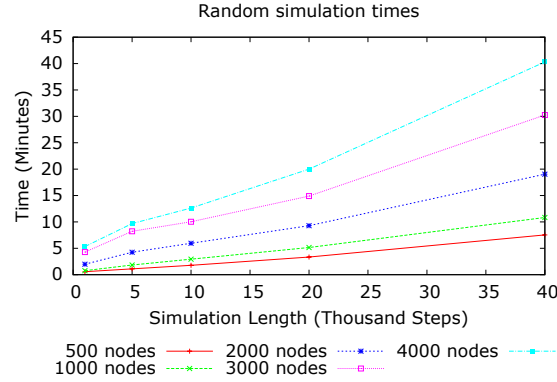


Figure 6.2: Random simulation times.

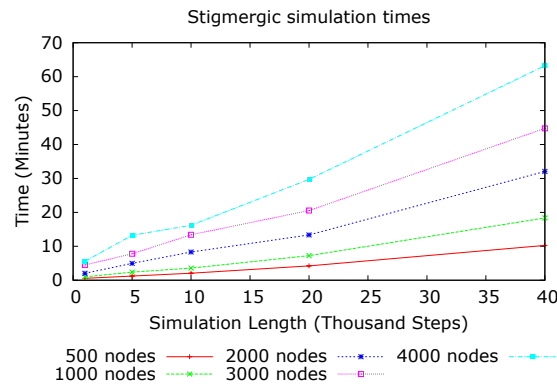


Figure 6.3: Stigmergic simulation times.

6.1.2 Memory Usage

Memory requirements also grow with the network size and simulation lengths (Figures 6.4 through 6.6). The memory usage is determined by the maximum value observed during the simulation execution, found in this case by querying the Java RunTime as in Listing 6.1.

6.1.3 Experimental Defaults

Due to the time and memory requirements of the simulations, a network size of 1,000 nodes is considered in this work. This provides an ample population of users with a range of experts and various routing options to create a suit-

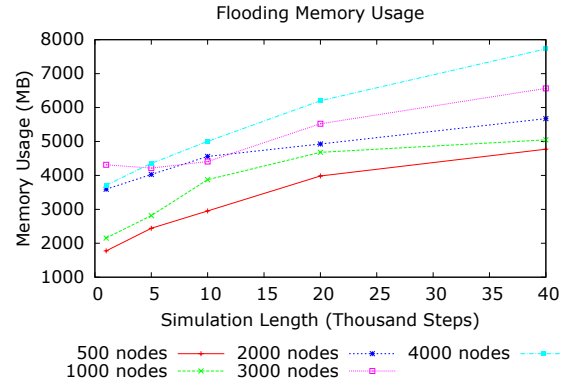


Figure 6.4: Flooding memory usage.

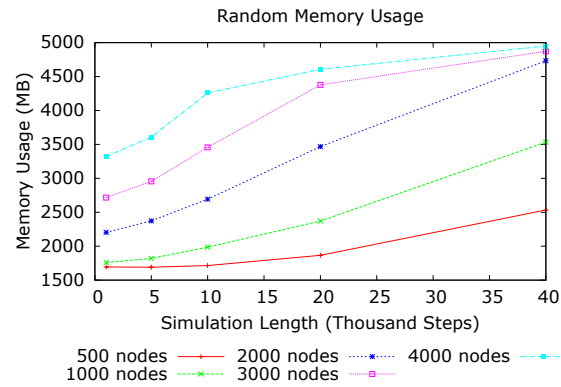


Figure 6.5: Random memory usage.

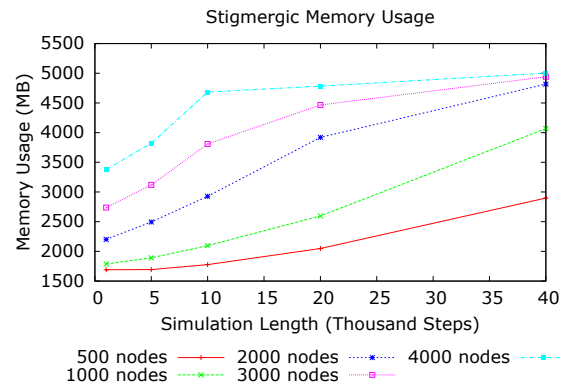


Figure 6.6: Stigmergic memory usage.

Listing 6.1: Memory Usage in Java

```
Runtime runtime = Runtime.getRuntime();  
memoryUsage = runtime.totalMemory() - runtime.freeMemory();
```

able routing challenge, providing a practical test bed for comparisons to be made. The flooding approach requires considerably more time and memory in comparison to the random and stigmergic approaches due to broadcasting messages throughout the network and the number of generated answers in response to a given question. As flooding is unacceptable in terms of required user attention and network load, it will only be used for comparison where absolutely necessary.

From this point onwards, variables are compared in turn, however, the setup of other parameters is based on findings found throughout this chapter. Some assumptions need to be made to allow for any comparisons to be made. Of particular interest is that simulations are typically compared using the C1 scenario presented in previous chapters, that being a scenario with users actively joining and leaving the network throughout the simulation lifetime.

6.2 Pheromones

The key feature of the stigmergic approach is the use of various pheromones to guide questions towards emergent experts. To understand the effects of the values chosen for each, this section provides a range of simulation results to identify the most sensible combinations. A great deal of investigation into pheromone values was conducted and can be seen for a large variety of scenarios in Appendix 8. The ratio between quality and attention provides a good means to identify the best combinations.

Here values are tuned to maximise the key metrics of interest, pushing up answer quality and controlling user attention. In a real-world implementation, global pheromone values may need to be tuned by extensive user feedback in the form of interviews and questionnaires. Due to the motivations of this work (specifically deniability) this process could potentially be a long and difficult task. Minor variations in user behaviour should not cause catastrophic performance problems with our routing approach. In a real-world implementation

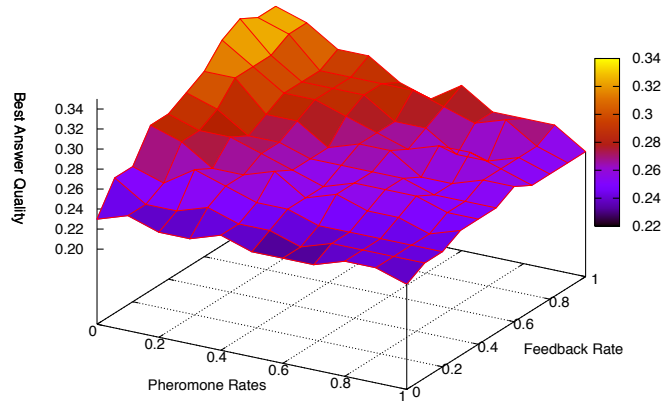


Figure 6.7: Best answer quality against phormone update values (constant).

it could also be possible to tune values locally based on user feedback and network performance.

6.2.1 Constant Increase Versus Proportional

The pheromones use constant rates as part of the pheromone update rule. One variation would be to consider some delta value (Δ) which updates pheromone strengths as a function of its current value. This section explores the differences between both methods. As shown in Figure 6.7 and 6.8, the constant approach will push the quality higher and faster, while the proportional approach is smoother and not as greedy, achieving a lower overall answer quality.

6.2.2 Warm up periods

The stigmergic routing takes time and user interactions to establish category related overlay networks. To correctly analyse the stigmergic approach an established network should be considered rather than a learning one. This can be achieved by only considering the results obtained after some suitable period of time. One such method of establishing a suitable and effective ‘warm up period’ is to make use of the Exponentially Weighted Mean Average (EWMA)

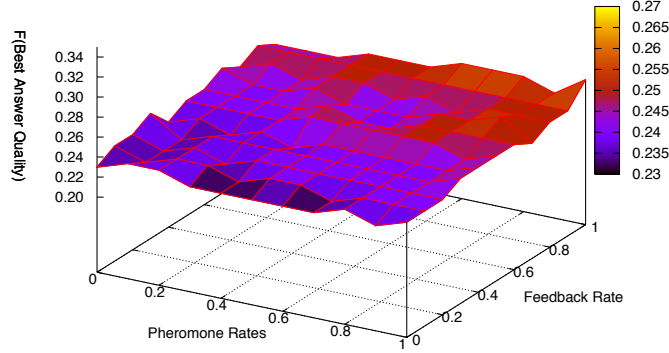


Figure 6.8: Best answer quality against pheromone update values (proportional).

[95], see equation 6.1¹ which can be used to see trends within noisy data. Within the Q&A networks users may ask and answer questions on various categories from different network locations. This creates an extremely noisy environment of answer qualities which can be overcome via the use of an EWMA with a low α value such as 0.01. Figure 6.9 provides the EWMA of network without churn showing how the average answer quality within the network improves over time above and beyond the random and flooding approaches. A dotted line identifies (Figure 6.9) a suggested warmup period to represent the point where the networking routing has been allowed to establish. Suitable warmup periods are selected later, taking into account the various churn scenarios.

$$EWMA_t = \alpha Y_t + (1 - \alpha)EWMA_{t-1} \text{ for } t = 1, 2, \dots, n.^2 \quad (6.1)$$

¹where: $EWMA_0$ is the mean of historical data (target), Y_t is the observation at time t , n is the number of observations to be monitored including $EWMA_0$, $0 < \alpha \leq 1$ is a constant that determines the depth of memory of the EWMA.

²where: $EWMA_0$ is the mean of historical data (target), Y_t is the observation at time t , n is the number of observations to be monitored including $EWMA_0$, $0 < \alpha \leq 1$ is a constant that determines the depth of memory of the EWMA.

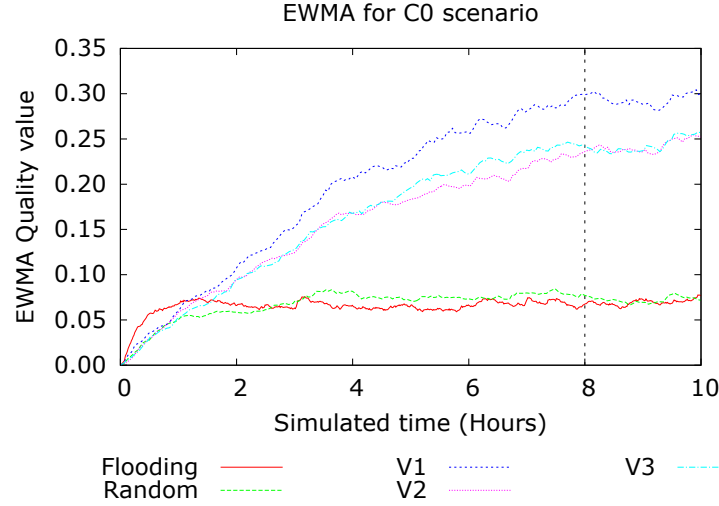


Figure 6.9: Exponentially weighted mean average answer quality.

6.2.3 Pheromone Rate, Feedback and Attention

Answers and user feedback trigger updates of local category pheromone values. As an answer flows through a link, the pheromone update value is used to adjust the local value. Feedback also triggers an update to take place with a constant value.

The routing can show greater greed towards experts by having a greater increase in pheromone values for feedback (indicating a good answer) in relation to the pheromone update value (any answer). By biasing feedback, the maximum levels (95th percentiles) of attention for a given user will increase (see Figure 6.11) as questions are pushed towards those members of the network with more expertise, increasing best answer quality (see Figure 6.10).

When using the *proportional* approach the increase in attention across the network is less significant (Figure 6.12 and 6.13). Overall the proportional approach performs poorly in the Q&A networks when compared to a constant increase and is subsequently ignored from this point.

6.2.4 Pheromone Maximums and Default levels

Existing stigmergic inspired work uses minimums and default pheromone levels to ensure that there is always some probability of selecting an active route (see Chapter 2), while setting maximums to avoid heavy bias towards any particular

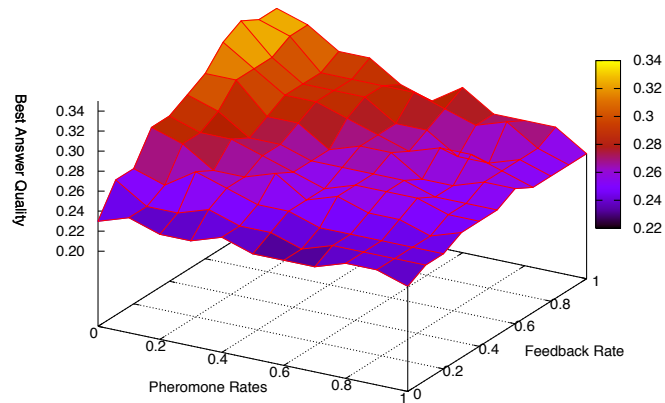


Figure 6.10: Best answer quality against pheromone update values (constant).

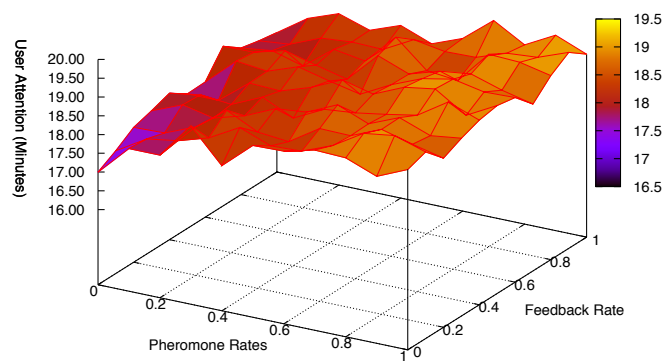


Figure 6.11: User attention against pheromone update values (constant).

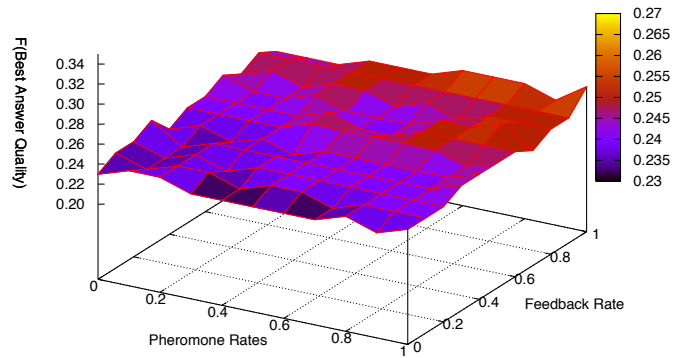


Figure 6.12: Best answer quality against pheromone update values (proportional).

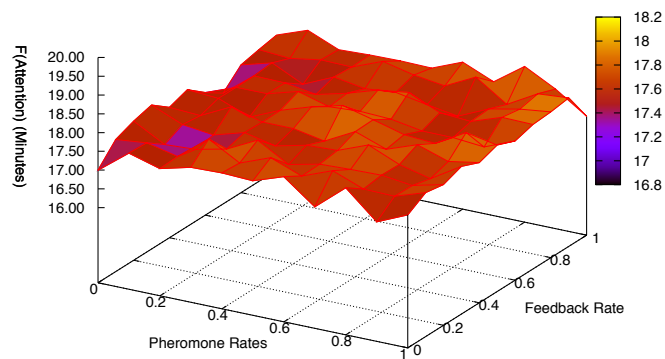


Figure 6.13: User attention against pheromone update values (proportional).

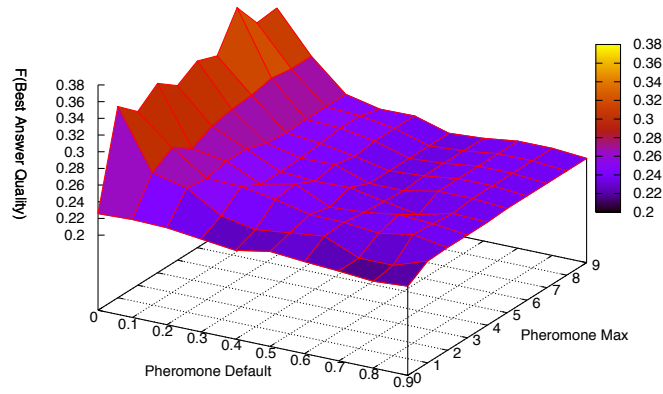


Figure 6.14: Default and maximum value effects on answer quality.

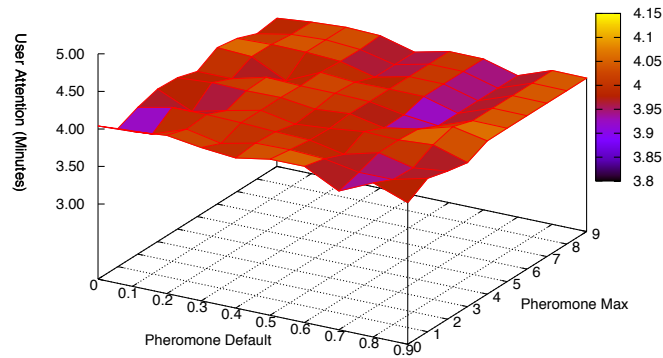


Figure 6.15: Default and maximum value effects on user attention.

link. Figure 6.14 shows the effect on quality by choosing different values for minimum and maximum pheromone scent levels on a given link. We can see a fairly even distribution of user attention in Figure 6.15 with varying minimum and maximum values. We can see that low defaults with higher maximum values work well.

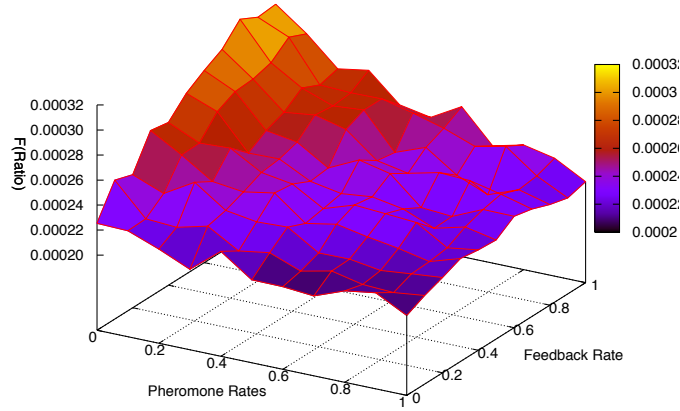


Figure 6.16: Ratio between quality and attention.

6.2.5 Balance of Quality and Attention

By using the constant pheromone strength increase technique there exists a ratio between quality and attention, see Figure 6.16. This ratio allows us to identify the maximum quality for the given user attention. The experiments show that using the preferred ratio between pheromone rates and feedback provides the highest ratio, namely feedback rate $>$ pheromone rate.

6.2.6 Load Balancing and Initial Values

A suitable pheromone reduction when forwarding a question through a link is considered here. V2 and V3 of the algorithm rely on a initial pheromone value for the loopback entry of the routing table relating to user interest categories. V3 affords load balancing by reducing the category pheromone strengths of those links which questions are sent through.

First, Figure 6.17 shows the variation in quality in V2 of the algorithm with various startup pheromone values for those categories users are initially interested in. Figure 6.18 shows how the variation of startup values reduces the level of overloading of nodes (bombarding) and Figure 6.19 shows a logarithmic scale plot for the percentage of unanswered questions.

Looking at the V3 algorithm we find some interesting properties. Figure 6.20 shows how its possible to boost the quality of answers with lower startup

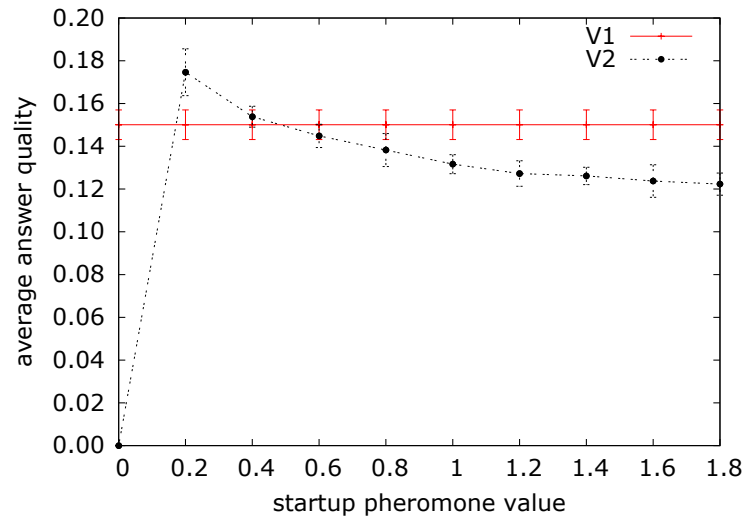


Figure 6.17: V2 pheromone startup value against answer quality.

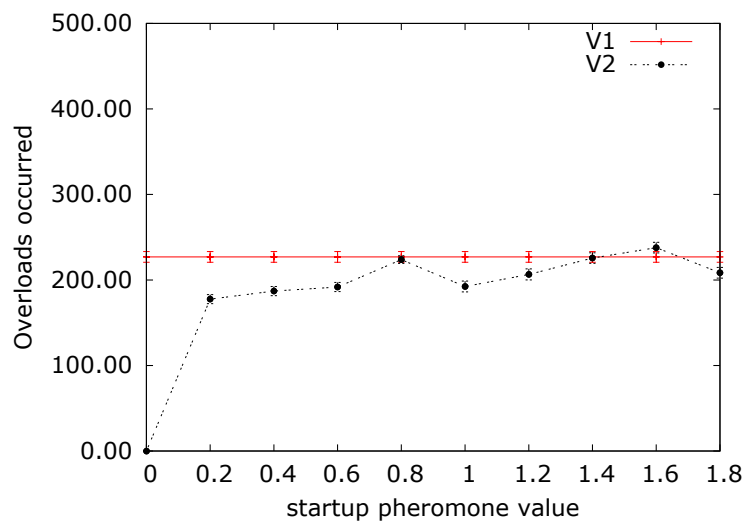


Figure 6.18: V2 pheromone startup value against bombardments.

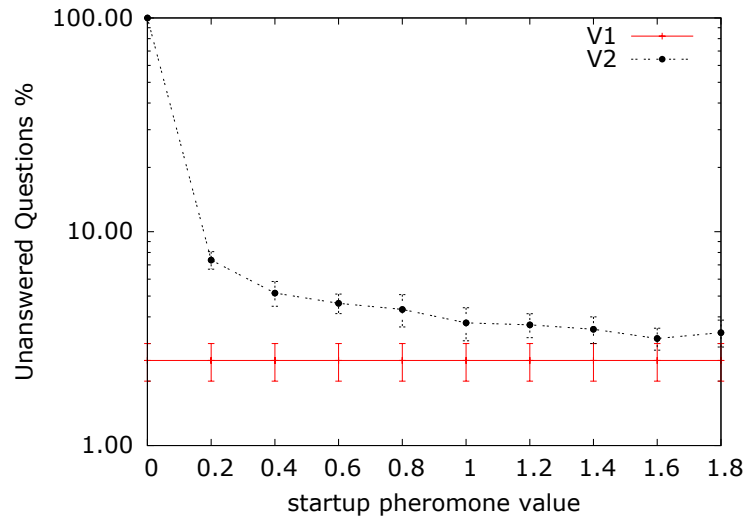


Figure 6.19: V2 pheromone startup value against unanswered questions.

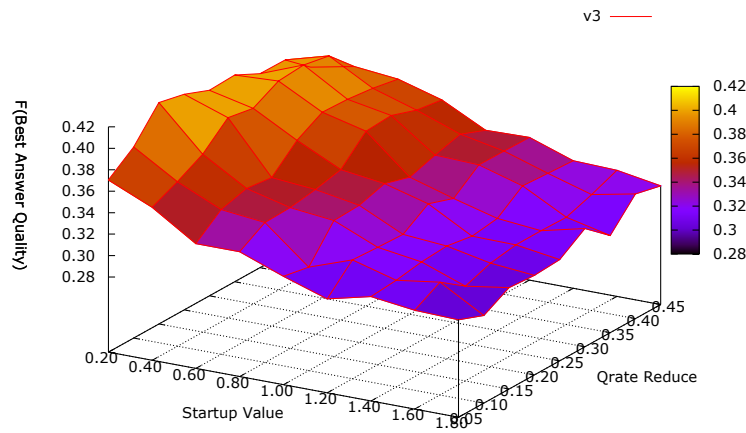


Figure 6.20: V3 pheromone defaults against answer quality.

values and higher reductions when forwarding questions. Choosing this preference will generate a higher proportion of unanswered questions (Figure 6.21). Of particular interest is that choosing higher values will reduce the number of answered questions, however it will also reduce the level of bombardments (Figure 6.22).

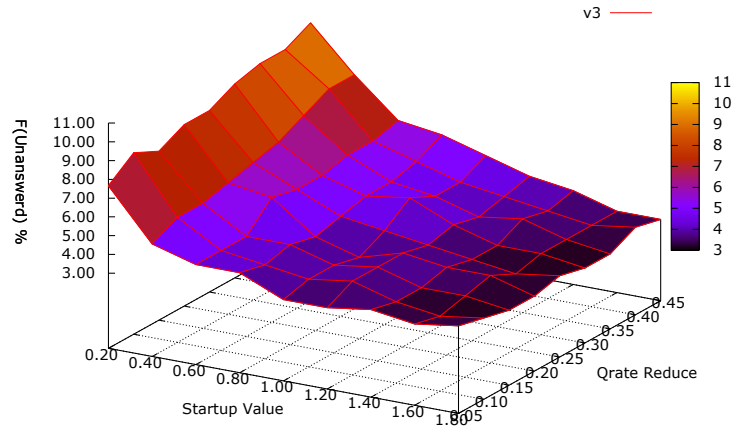


Figure 6.21: V3 pheromone defaults against unanswered questions.

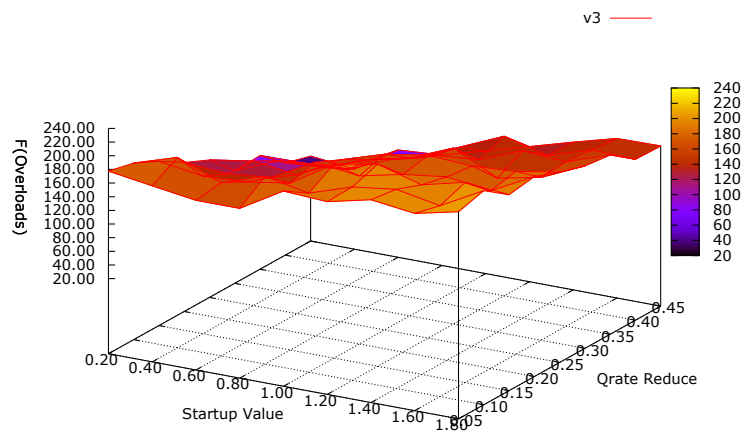


Figure 6.22: V3 pheromone defaults against overloads.

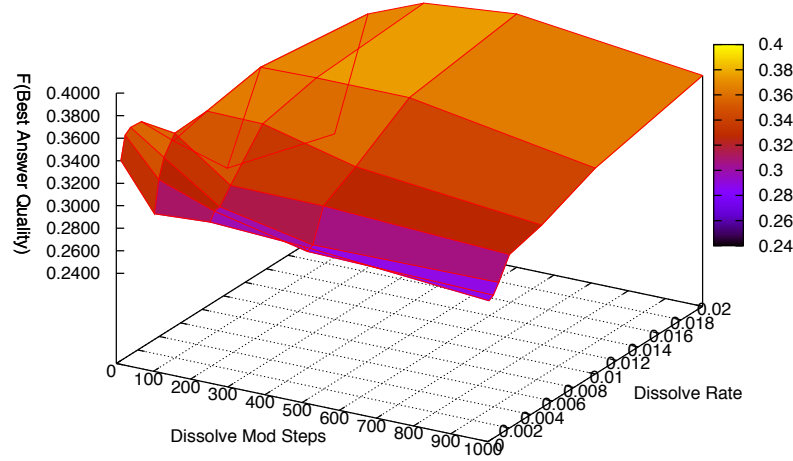


Figure 6.23: Pheromone evaporation against answer quality.

6.2.7 Pheromone Evaporation

Pheromone scent levels are reduced by a constant amount (dissolve rate) at regular intervals (mod steps). Allowing pheromones to evaporate provides a form of negative feedback in the routing. Figure 6.23 provides results of various pheromone evaporation rates in identical networks showing that the quality can be pushed up by reducing pheromone scent levels in larger intervals at larger rates. The levels of attention consumed on average by the users is fairly constant (Figure 6.24).

6.2.8 Adjustment of parameter choices

To summarise, using low (< 0.2) pheromone update values (when an answer is received) with high (> 0.6) feedback levels (good answers) will achieve the highest ratio between quality and attention (Figure 6.16). A suitable evaporation rate (≈ 0.01) and interval (≈ 500 steps) will be used. V2 and V3 of the algorithm should choose low startup values ≈ 0.2 and V3 should choose a sensible decrease such as ≈ 0.25 .

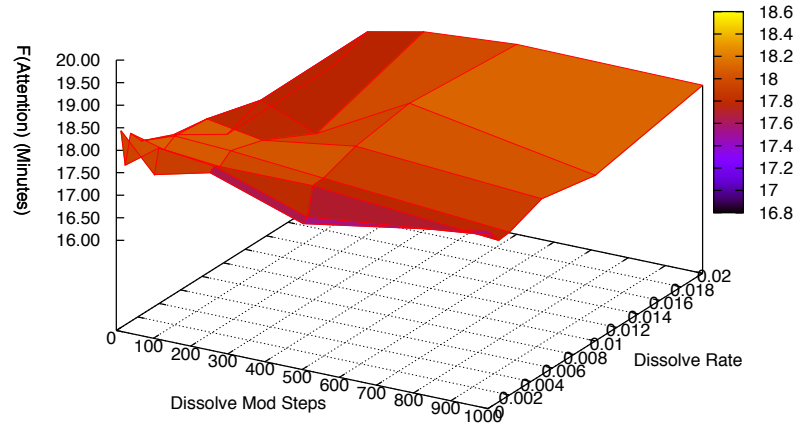


Figure 6.24: Pheromone evaporation against user attention.

6.3 Generic Protocol

The underlying protocol requirements (e.g. single hop routing) determines many of the features of the routing strategy. This section evaluates the key aspects and how they adjust the routing outcomes.

6.3.1 Network Properties

Investigating networks of 1,000 nodes will allow reproducibility of exact network structures and conditions via master seeding. Table 6.1 presented several pieces of information regarding the initial static networks. The distribution of local degrees are calculated via the Erdős and Rényi constant k discussed in Chapter 4. Figure 6.25 provides a visual representation of the network, albeit a cluttered and difficult structure to interpret visually.

6.3.2 Question Time-To-Live Values

The Time to Live (TTL) values determine how deep into the network questions may propagate, and as such, sensible TTL values should be selected. This

Variable	Value
Nodes	1,000
Edges	7,619
Avg Degree	15.238
Network Diameter	4
Avg Path Length	2.805
Degree Power Law	24.338
Avg Clustering Coefficient	0.014

Table 6.1: Initial network characteristics.

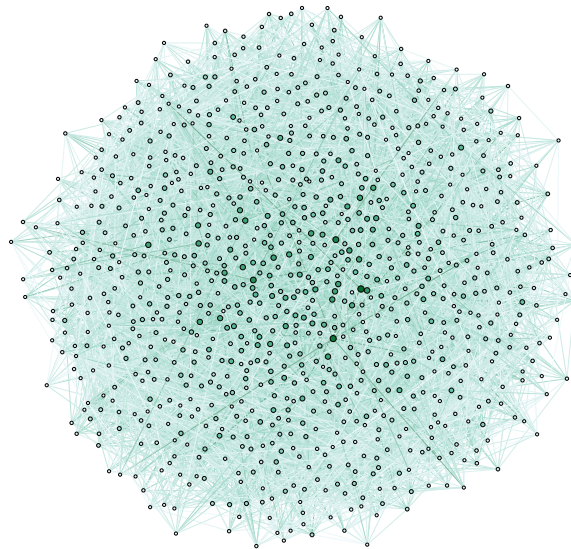


Figure 6.25: Network visualisation of 1000 node random network.

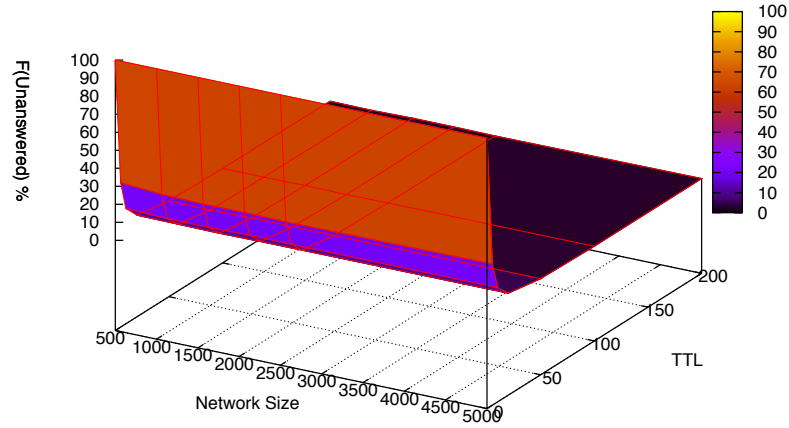


Figure 6.26: (Stigmergic) TTL values against unanswered questions.

section presents the simulation results with a range of TTL values and network sizes to provide an image of how they affect simulations, shown in Figure 6.26. It is possible to have slightly lower TTL values for the stigmergic approach in comparison to random hops, in addition the performance appears to flatten out around 100 hops. Random TTL results can be seen in Figures 6.29 through 6.31³.

Figure 6.27 provides results of the change in quality created by a range of TTL values in various network sizes. The levels of attention can also be seen in Figure 6.28.

6.3.3 Number of Answers Required

The number of answers required for each question will directly determine the possible levels of user attention consumed per question. With a high answer requirement the random and stigmergic approaches will mimic flooding, however with lower requirements it becomes harder to locate experts by chance.

³The flooding technique attempts to reach all nodes in the network while ignoring any TTL, and is therefore excluded.

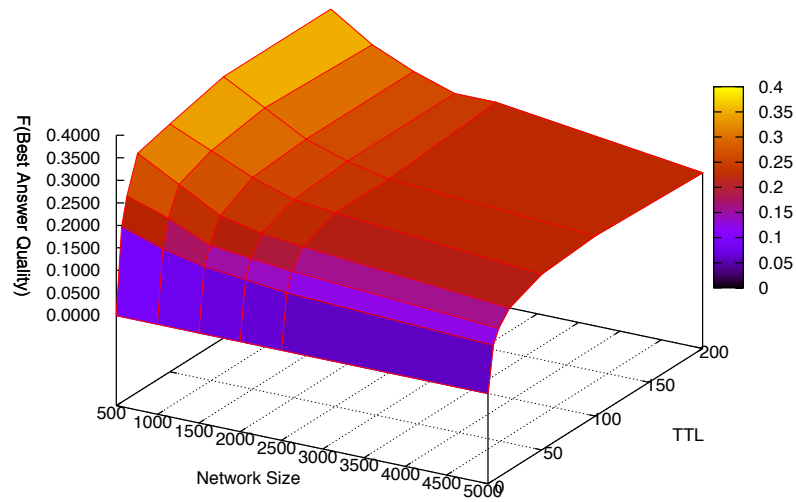


Figure 6.27: (Stigmergic) TTL values against answer quality.

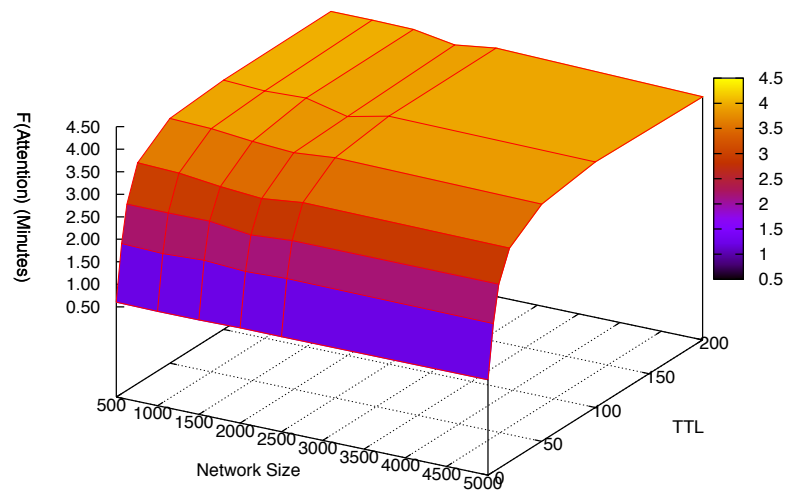


Figure 6.28: (Stigmergic) TTL values against user attention.

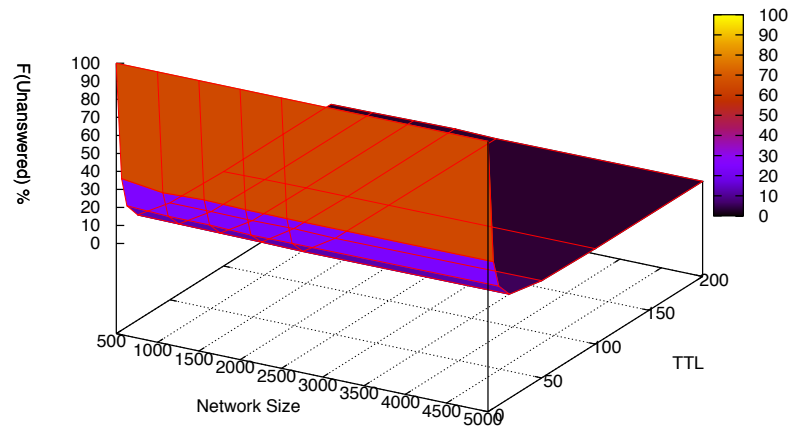


Figure 6.29: (Random) TTL values against unanswered questions.

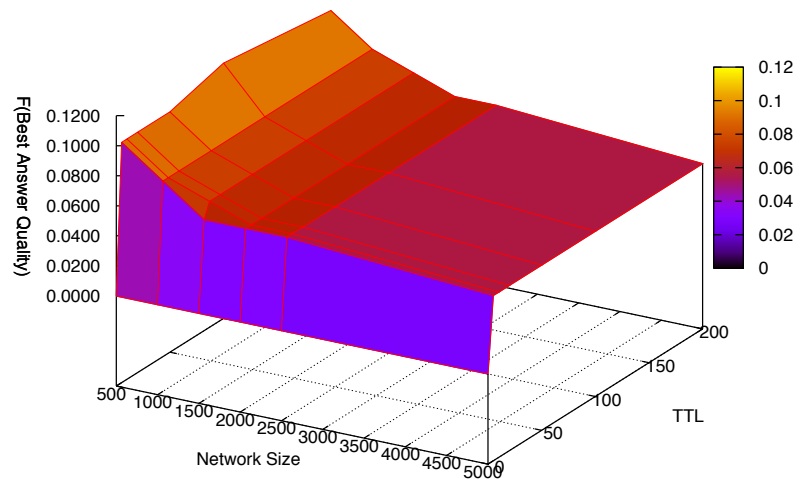


Figure 6.30: (Random) TTL values against answer quality.

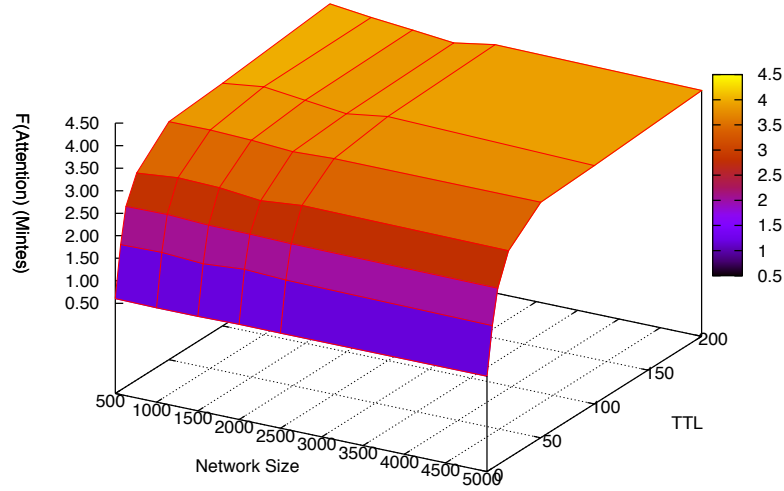


Figure 6.31: (Random) TTL values against user attention.

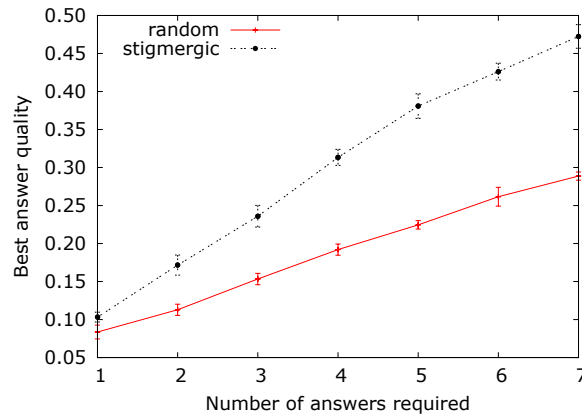
As can be seen in Figure 6.32 the answer quality improves with the number of answers required from both the stigmergic and random approaches – however the stigmergic method always leads the way. The attention requirements rise fairly rapidly with the number of answers requested. Requiring a handful of answers would seem appropriate (supplying a balance between quality and attention) for use with comparing approaches.

6.4 Summary of the Generic Protocol

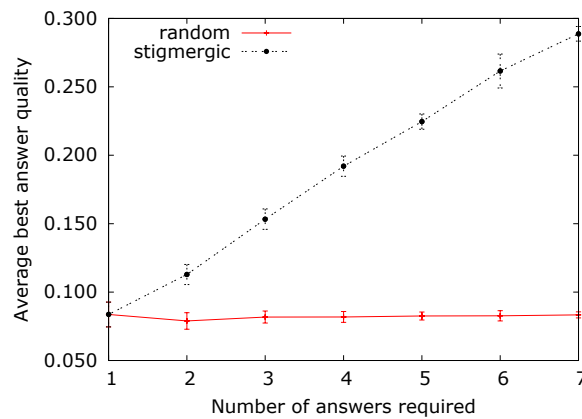
Regardless of network size, we need a suitably high number of hops (TTL) for each question, somewhere around 100 hops for the stigmergic approach (Figures 6.26 through 6.28).

We should request a suitable number of answers from the network to get better improvements in quality. The levels of user attention consumed is proportional to the number of answers required for each question. As the attention levels consumed are very similar for both routing approaches it is possible to choose the levels of attention to consume to serve a particular question. With the stigmergic approach leading in terms of quality (Figure 6.32) no matter

Number of answers against best answer quality.



Number of answers against average best quality.



Number of answers against average user attention.

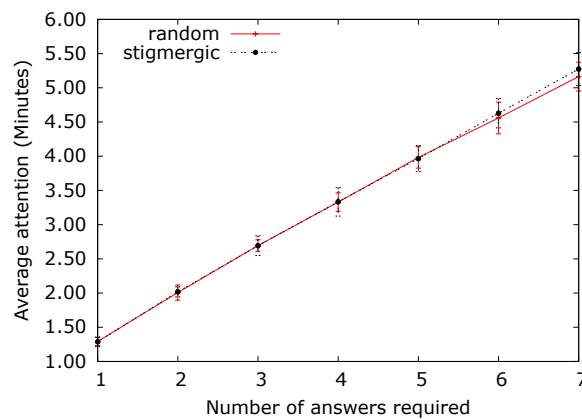


Figure 6.32: Number of answers required consequences.

how many answers are required it would seem appropriate to pick a value which gives a significant improvement, such as requesting 5 answers for each question.

6.5 User Model

Users are generated directly from the discrete data distributions observed in Chapter 2. There are however several configurable elements of the simulated users which are worthy of exploration. This section aims to provide a clear presentation of the various effects caused by these configurable members.

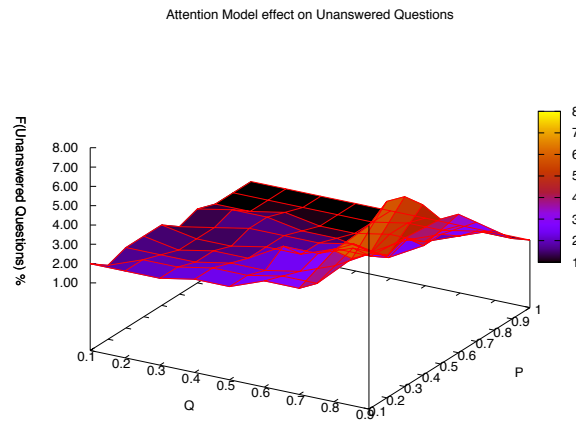
6.5.1 Transitional Probabilities

Users follow the Markovian model as presented in section 3.4.2. The values which define the probabilities of a user paying attention to the system can be defined by two values (P & Q), where users will transition between or remain in one of the possible states. This section provides an evaluation of the effects caused by various attention models.

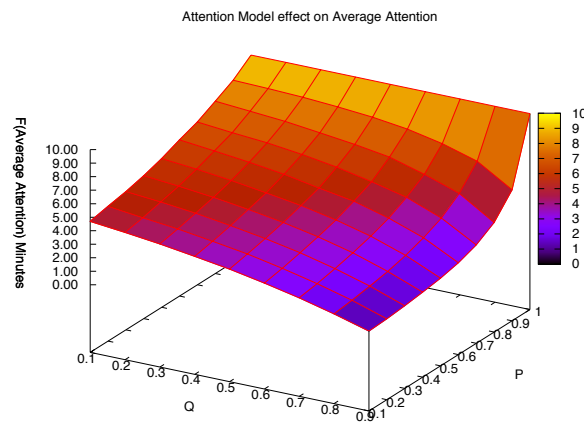
The users transition between two states: 1) paying attention and actively participating in the system and 2) being idle where they are not available or performing any user related tasks such as answering questions (however still supporting routing at the network level). Users may transition between states probabilistically at each time step in the simulation.

Users begin in the paying attention state and remain at each step with probability P and $1 - P$ of transitioning into the idle state, then with a probability of Q of remaining idle and $1 - Q$ of becoming active once again. We can explore the effects of P and Q in terms of the proportion of unanswered questions, user attention and answer quality. Figure 6.33 demonstrates the effects over possible values of P and Q in relation to the number of unanswered questions, user attention and best answer quality within identical networks. Interestingly, as users pay greater attention to the system they also generate more questions. Intuitively, while users participate less in the system a greater proportion of questions go unanswered.

P and Q effect on unanswered questions.



P and Q effect on attention.



P and Q effect on quality.

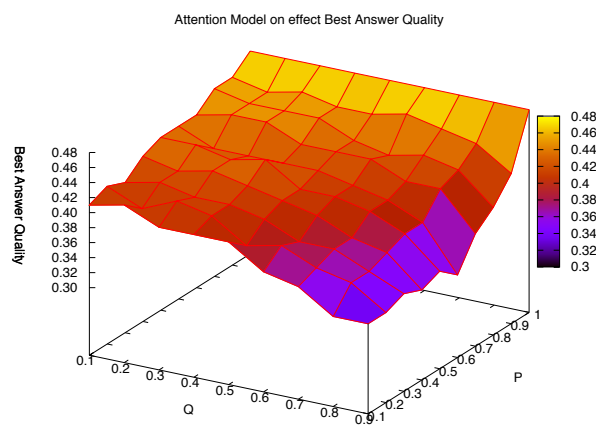


Figure 6.33: User model effects on best answer quality and user attention.

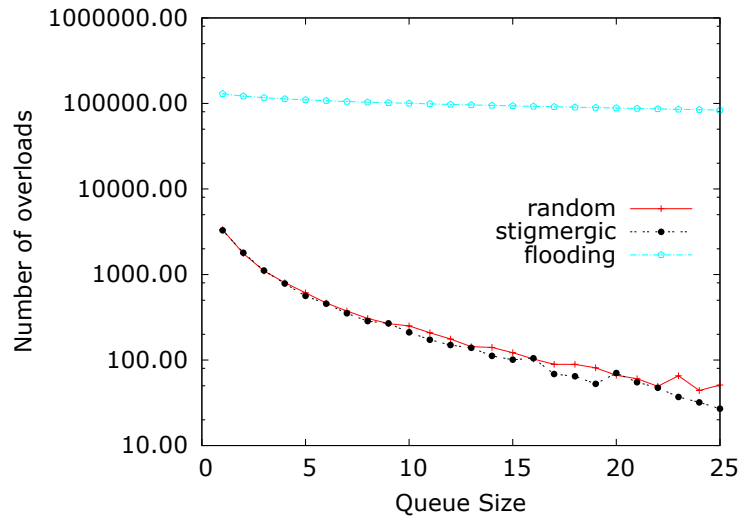


Figure 6.34: Queue sizes causing overloading.

6.5.2 Priority Queue Size

Each user has a fixed size priority queue in which questions of interest are stored waiting for user attention to generate answers. It is interesting to consider the effect of queue size on the system and results are presented in Figure 6.34 with a logarithmic y axis. Smaller queue sizes will cause local overloading. Overloading can be overcome by setting an appropriate local queue size. If too many questions are asked and local queues are insufficient in size, overloading will occur in all approaches. As can be seen, the flooding approach performs poorly in this area.

6.5.3 Question-Asking Rate

The questions generated in the network cause a greater load on the network and more demand and participation from the networked group of users. This section explores the number of questions which can be realistically dealt with. Figure 6.35 explores an increasing question asking probability, resulting answer quality (Figure 6.35) and the required user attention (Figure 6.35). The quality of the stigmergic approach improves as the quantity of interactions grow in the network. As the question rate increases so does the network traffic and the levels of overloading (when queues become full) taking place in the system. Flooding suffers badly from overloading as questions rates increase.

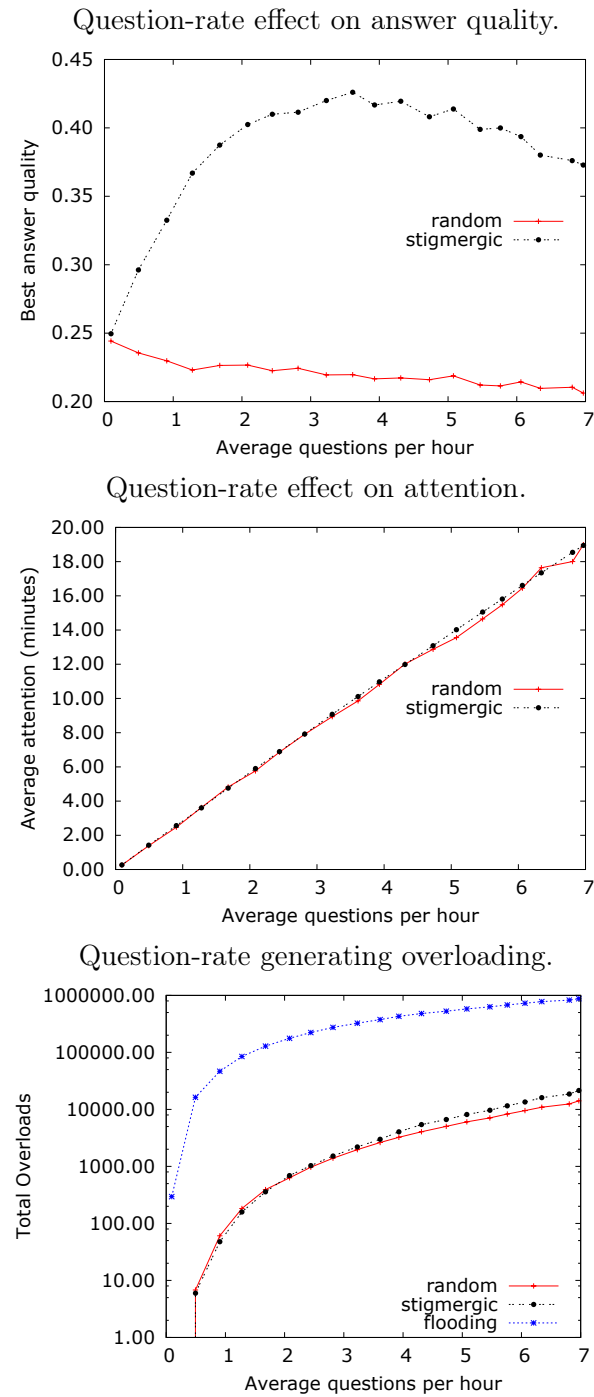


Figure 6.35: Question asking rate consequences

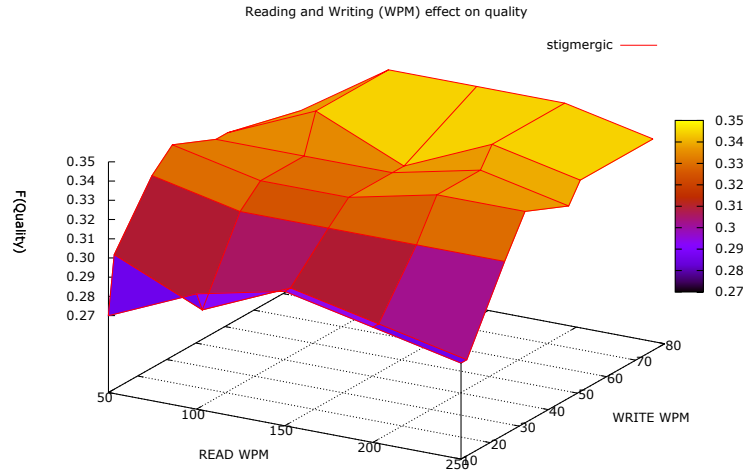


Figure 6.36: Reading and writing abilities against mean best answer quality.

6.5.4 Reading and Writing Abilities

As presented in Chapter 3, users ability to compose and read text in digital environments is determined by the type of display and input devices available. This section examines how words per minute (WPM) reading and writing speeds dictate the ability for users to generate questions and answers within the Q&A networks. Figure 6.36, 6.37 and 6.38 present a range of reading and writing rates and how they effect key system metrics.

As long as the users have the ability to compose and read questions and answers at a responsible speed, there is a small impact on the quality or proportion of answered questions. As user abilities reduce, it requires more overall attention (Figure 6.38) to deal with questions and create answers due to having to spend longer periods of time serving each request – as a result more questions are left unanswered (Figure 6.37).

6.6 Summary of User Model Evaluation

We have evaluated the effects of various attention models and the way in which attention and quality are adjusted. We cannot assume 100% from the network users ($P=1$), however its possible to choose a P and Q value < 1 to achieve good results without assuming full attention. Fortunately the routing is able

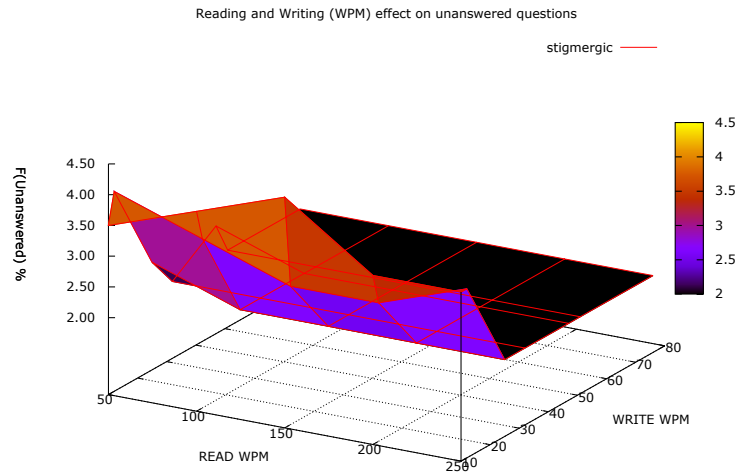


Figure 6.37: Reading and writing abilities against unanswered question.

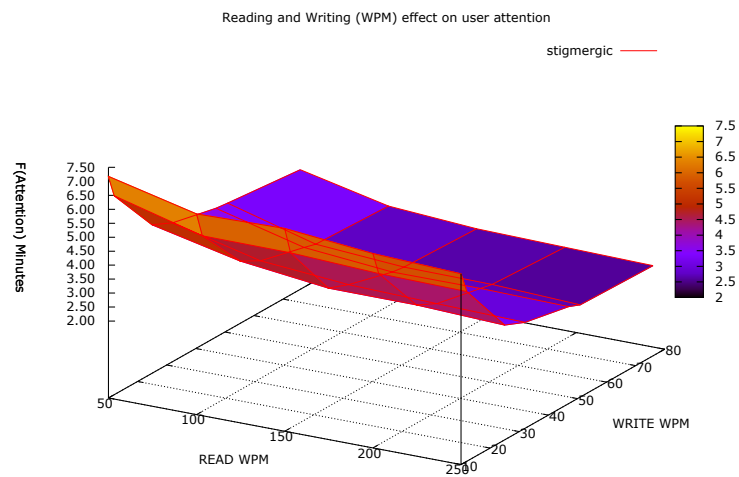


Figure 6.38: Reading and writing abilities against user attention.

to produce good results across the various attention model ranges, improving when users pay greater attention.

Users will need the ability to read and write questions and answers at reasonable rates, with existing research pointing towards 19 WPM writing and 250 WPM reading, it would seem an acceptable choice of values.

A suitable question asking rate should be used, where we know that the attention requirements and quality will increase as more user activity participates. Users are unlikely to consistently ask many questions per hour, but approximately one question per hour could be anticipated. Suitable local queue sizes should be chosen to avoid overloading, with a queue size of around 15 the level of overloading can be reduced.

6.7 Network Churn

The simulated environment becomes more interesting and realistic when churn is added to model the joining and leaving of network nodes. This section explores the effects that churn has on the system's key metrics using a variety of possible variations. We find that the key metrics are affected by λ and k of the Weibull distribution used – such that as users stay for longer periods of time the routing is able to learn about the expertise in the network and therefore increase answer quality and in turn as users begin to be used more, the attention requirements will rise.

We can evaluate the various churn scenarios (C0, C1, C2) to choose suitable warm-up periods. Figures 6.39, 6.40 and 6.41 provide the EWMA's for C0, C1 and C2 scenarios, with warm-up periods presented as dotted lines. The warm-up periods give a point during the simulation when the quality flattens out and hence the warmup periods are well approximated.

6.7.1 Answer Quality

The longer users remain connected the network, the greater the opportunity for the routing to learn. Figure 6.42 provides a view of the effects of churn which are directly determined by the Weibull distribution parameters. The quality is improved for longer session durations, created by higher lambda values in the underlying Weibull distribution.

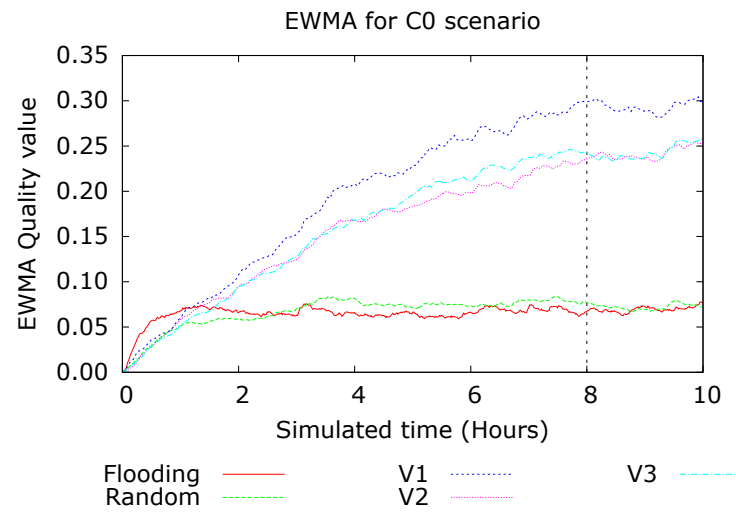


Figure 6.39: Exponentially mean weighted average (C1).

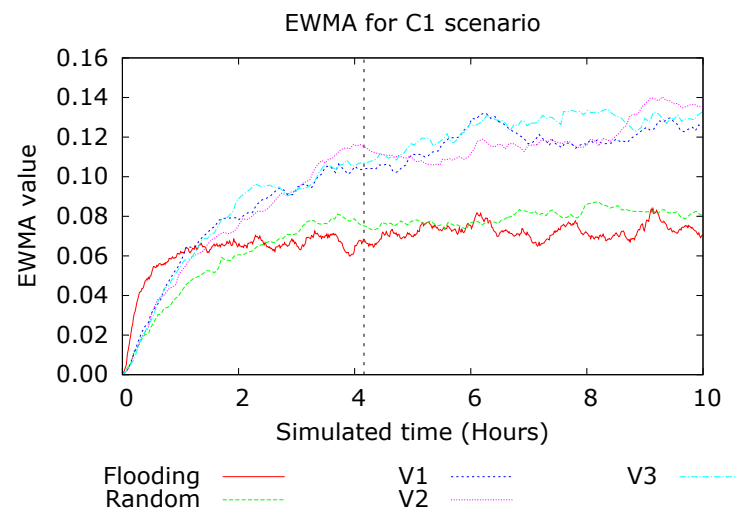


Figure 6.40: Exponentially mean weighted average (C1).

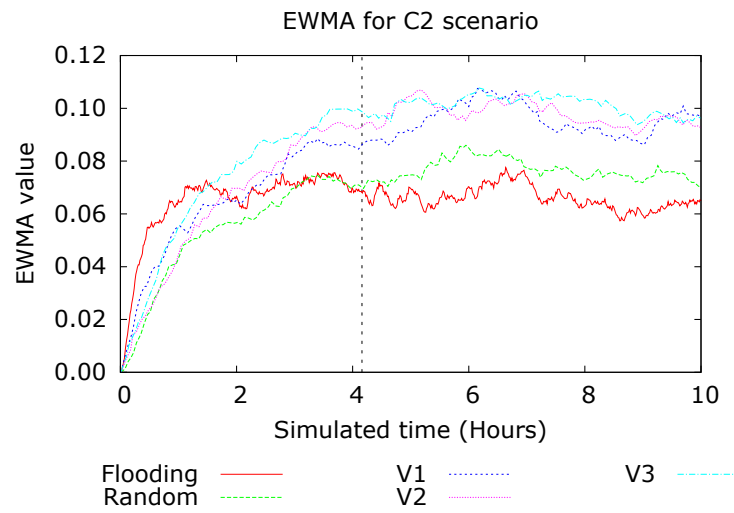


Figure 6.41: Exponentially mean weighted average (C2).

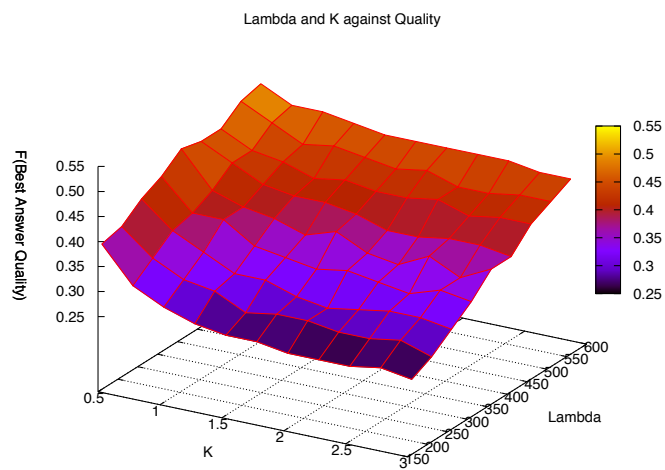


Figure 6.42: Churn effects on answer quality.

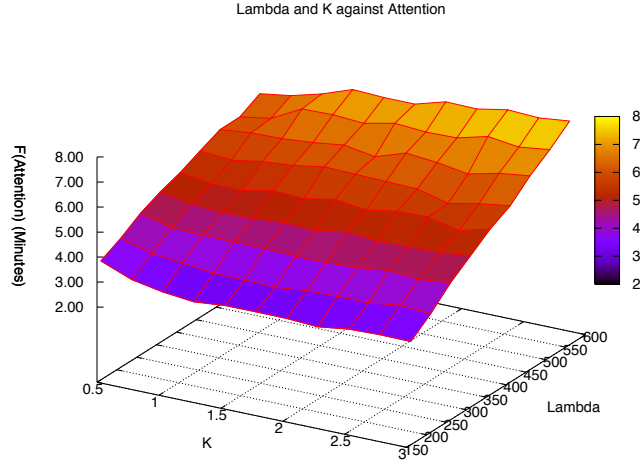


Figure 6.43: Churn effect on user attention.

6.7.2 Attention

Figure 6.43 provides results for the effects of churn on user attention. As users stay for longer periods of time there is a greater opportunity to utilise them as a resource and therefore the attention consumed will increase.

6.8 Scalability

Much larger networks of 10,000 nodes with the C1 churn scenario have been simulated. These larger networks take considerably longer to simulate than 1,000 nodes. The stigmergic approach is still able to keep the relative comparable improvements from the random hops base line approach. Within larger network simulations we are able to clearly see the improvement in the exponentially weighted mean average quality over time in Figure 6.47 with V2 and V3 performing the best. Within larger networks the quality calculations include the full set of possible experts and therefore all results are condensed, as questions are serviced more easily when a larger collection of nodes are held together. Possible experts included in our answer quality metric may be deep within the network and practically unreachable before questions are consumed and answered by interested users. We are still able to see a clear improvement however in best answer (Figure 6.44) and average answer (Figure 6.45) quality,

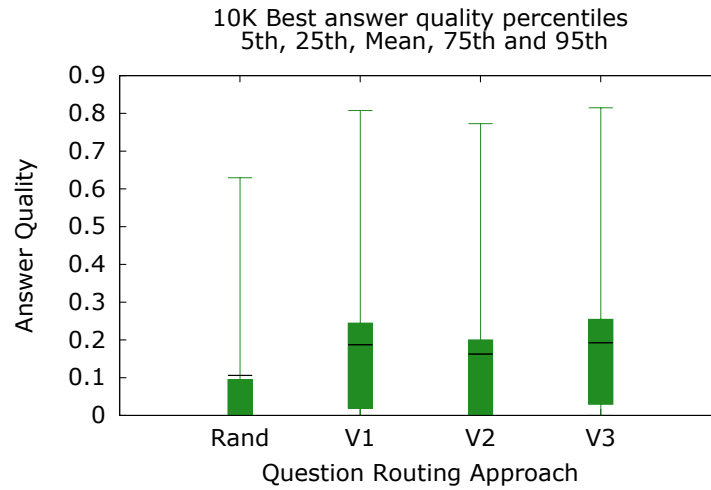


Figure 6.44: Best answer quality.

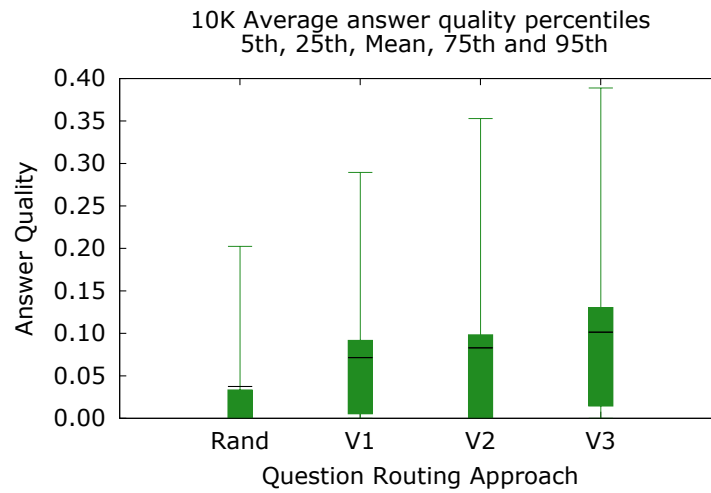


Figure 6.45: Average answer quality.

still with comparable attention (Figure 6.46) requirements for all stigmergic based approaches.

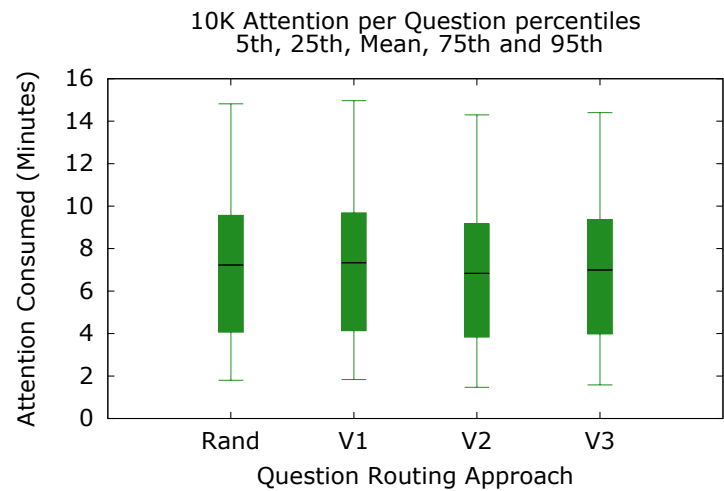


Figure 6.46: Attention per question.

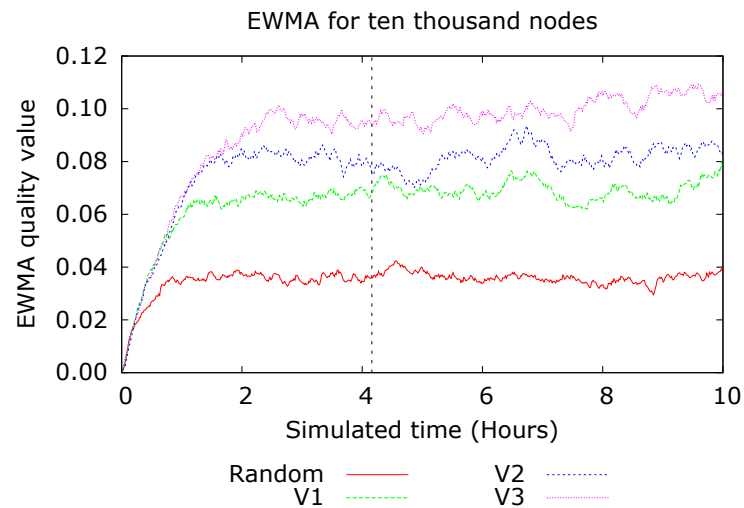


Figure 6.47: EWMA quality.

Variable	C0	C1	C2
Weibull λ	–	269.79	89.97
Weibull k	–	3.07	1.13
Simulation Length	36,000		
Network Size	1,000		
Question Asking Rate	0.000175		
Local Queue Size	15		
Attention model Q	0.9996		
Attention model P	0.9996		
WPM Read/Write	250 / 19		
Default Pheromone Value	0.06		
Answer Increase	0.05		
Feedback Increase	0.8		
V3 Load Balance Decrease	0.25		
V2/3 Initial Value	0.2		
Warmup Period	15,000 steps		
Answers Requested	5		
Evaporation Rate	0.02		
Evaporation Interval	400 steps		

Figure 6.48: Simulation parameters.

6.9 Results

With a suitable evaluation and inspection of the many simulation variables, routing parameters and user model, it is possible to evaluate and compare the stigmergic question routing approaches in realistic settings. The parameters and configuration used in these simulations are presented in Table 6.48.

Ten different seeds, and hence network and question conditions, are tested. Results are presented using percentiles (5^{th} , 25^{th} , *mean*, 75^{th} , 95^{th}) due to the variation in the number of questions, network set up and answers based on random seeding across runs. The n^{th} iteration of each approach has an identical seed to the n^{th} iteration of other approaches and is therefore a like-for-like comparison.

6.9.1 Best Answer Quality

Figure 6.49 shows the percentiles of best answer quality across each approach for each churn scenario. Flooding performs the best with high quality across the board, and random in the final position not surpassing the stigmergic ap-

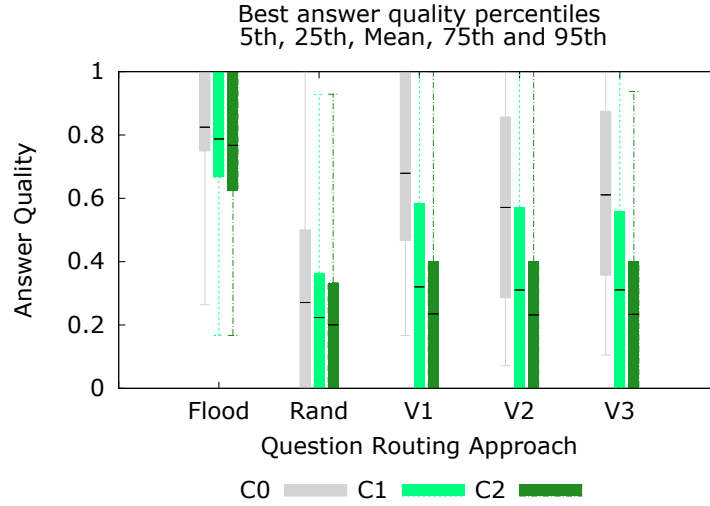


Figure 6.49: Best answer quality results.

proaches. V1, V2 and V3 of the stigmergic algorithm are each able to improve on the random routing approach, each reducing in performance as churn increases. Churn plays an important role in answer quality for all approaches, where the higher the level of churn the greater the reduction in answer quality.

6.9.2 Average Answer Quality

The average quality of the total pool of answers received per question is considerably improved with the stigmergic approaches. In Figure 6.50 the average answer quality is considerably better than flooding and random for each variation of the approach.

6.9.3 Path Lengths and Network Load

Figure 6.51 presents the *total* number of hops generated by a particular question. In this case the combined total of hops generated by requesting 5 answers per question. The path lengths of the V1 main algorithm are reduced in comparison to random. The V2 and V3 approaches require considerably more hops to fulfil requests as nodes learn about the author's expertise. The huge network load generated by flooding (off the scale) is particularly apparent here.

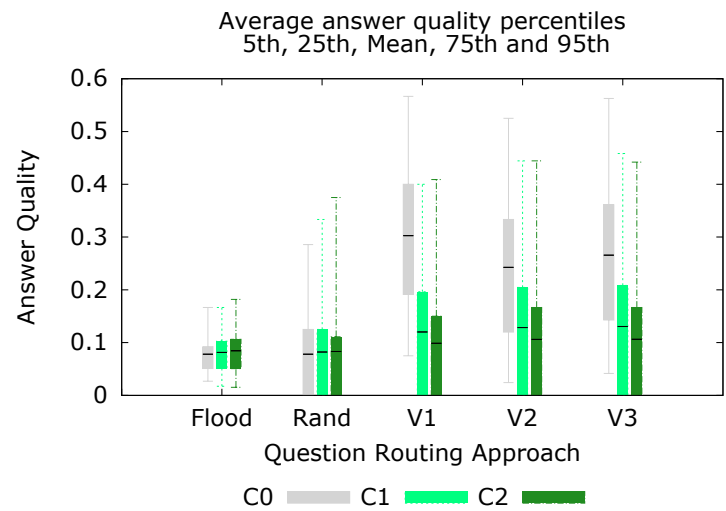


Figure 6.50: Average answer quality results.

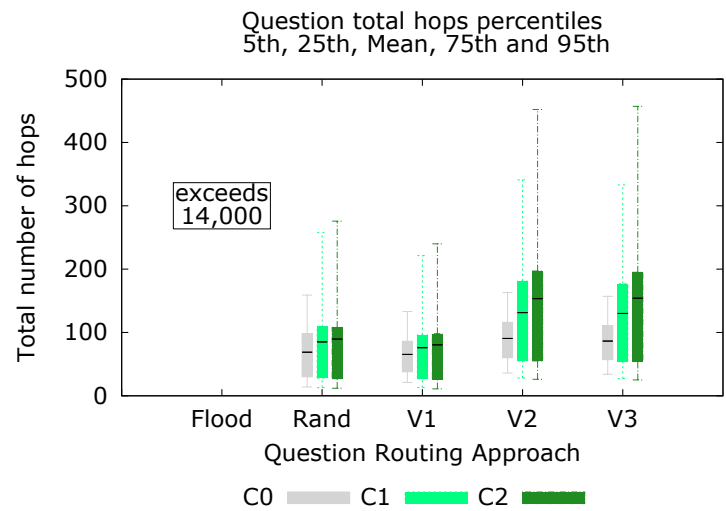


Figure 6.51: Network load results.

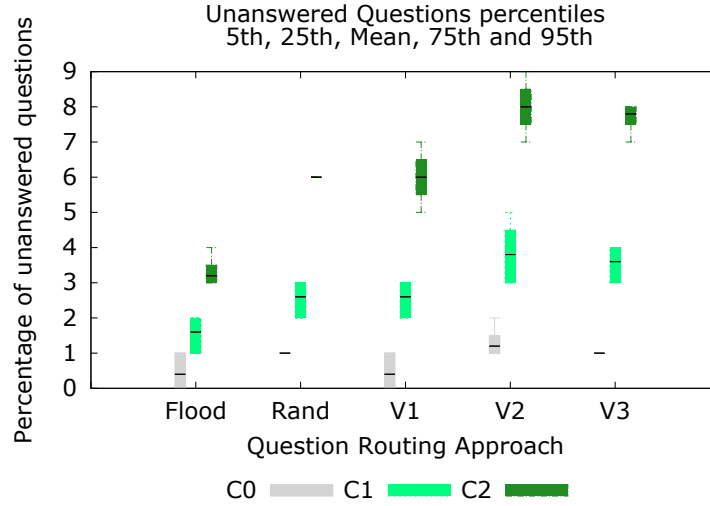


Figure 6.52: Unanswered questions results.

6.9.4 Unanswered Questions

In Figure 6.52 the percentiles of unanswered questions are presented. The flooding approach produces the least unanswered questions (those being questions asked without a single answer being delivered to the author). The main V1 stigmergic approach outperforms random with V2 and V3 performing slightly worse but with all approaches achieving under 10% unanswered questions.

This suggests that getting *any* answer to a question is not a major concern. It is the quality and attention that are the key metrics of interest.

6.9.5 Consumed User Attention

The attention consumed by the users of the network is of particular interest; the levels need to be in comparison with the fair random routing to promote a fair distribution of user effort.

6.9.5.1 User Attention Per Question

For each question the levels of attention consumed are recorded. Figure 6.53 presents the levels of attention consumed per question in percentiles for each algorithm and churn scenario. The levels of attention consumed with the flood-

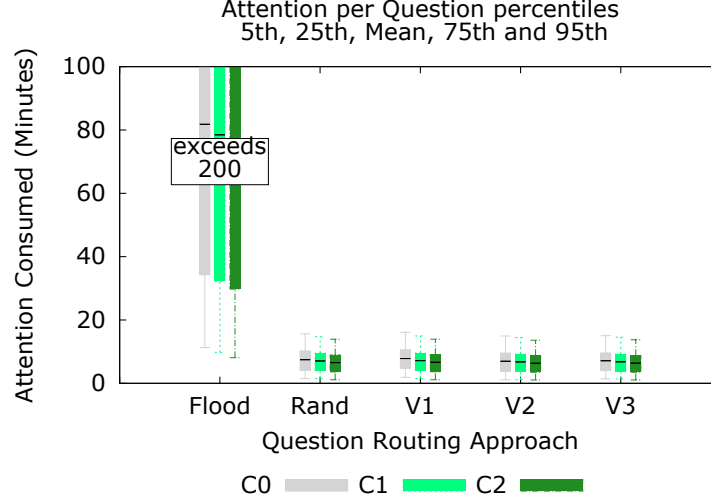


Figure 6.53: Attention per question results.

ing approach is unacceptable, however the stigmergic approaches are far more inline with random while exhibiting higher 95th percentiles. This is due to experts being forwarded more relevant questions to answer and hence contributing more attention from the routing protocol. This effect is particularly apparent in the C0 (churn-less) scenario where the network is given a long time to learn about the user expertise available and establish many pathways towards them. The V2 and V3 approaches reduce the 95th percentile while keeping similar quality benefits (Figure 6.49 and 6.50).

6.9.5.2 Total Attention Per User

Attention levels consumed per user are presented in Figure 6.54. Again we can see the unacceptable levels generated by the flooding approach. The stigmergic approaches are much more fair and inline with the random approach, while having an increased 95th which again is eased with the V2 and V3 approaches of the algorithm. With the C0 churn scenario the 95th percentiles are noticeably higher as the network is successfully established.

6.9.6 Exponentially Mean Weighted Averages

Evaluating the EWMA for each churn scenario is also of interest to see the clear advantage in average answer quality over time. The warmup periods

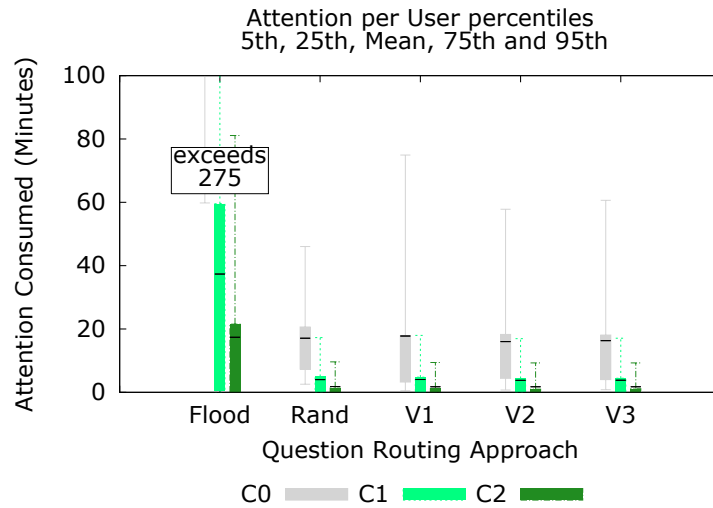


Figure 6.54: Attention per user results.

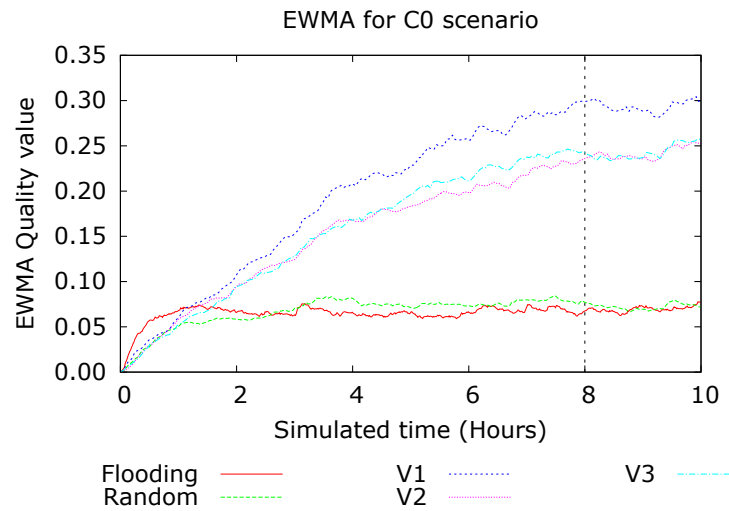


Figure 6.55: Exponentially mean weighted average (C0).

are included to signify the proportion of results which are removed from the results. Example EWMAs are presented for scenarios C0 (Figure 6.55), C1 (Figure 6.56) and finally C2 (Figure 6.57).

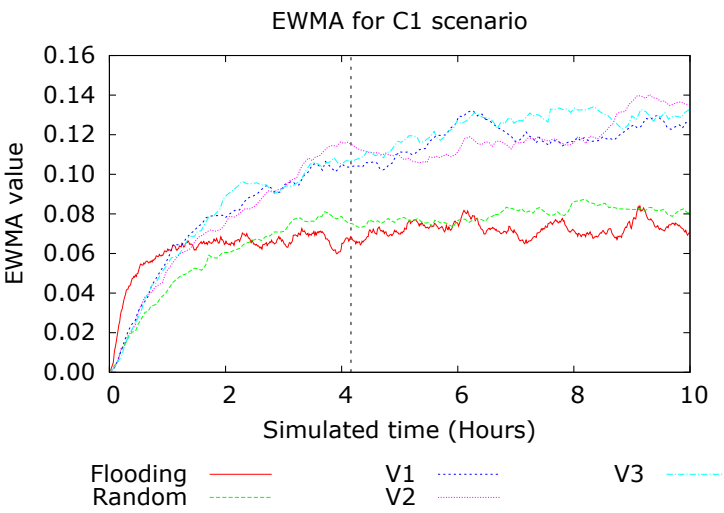


Figure 6.56: Exponentially mean weighted average (C1).

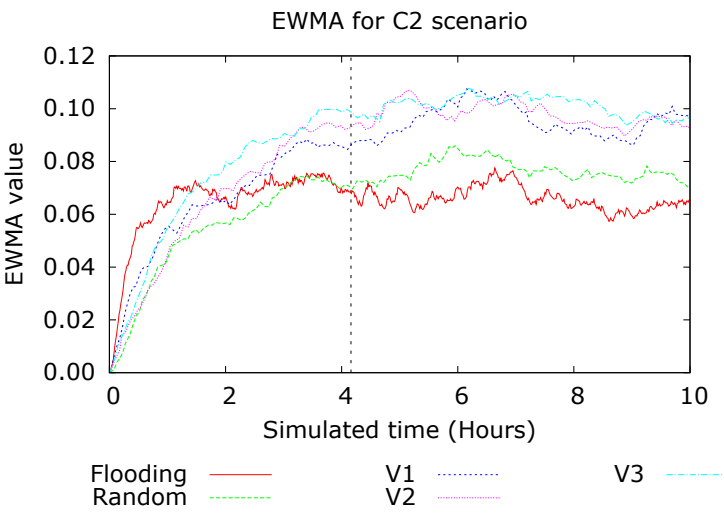


Figure 6.57: Exponentially mean weighted average (C2).

6.9.7 Summary

With a suitably configured stigmergic routing approach the quality of responses can be significantly improved in comparison to the best quality of a random approach and the average of flooding and random (Figures 6.49 and 6.50). The distribution of attention reduced and controlled (Figures 6.54 and 6.53) with acceptable path lengths (Figure 6.51) and proportion of answered questions (Figure 6.52). From this section of results, in a churn free environment, the stigmergic approach will learn routes towards the more knowledgeable members of the network, while distributing the load accordingly with the loopback orientated approaches V2 and V3. In more demanding churn scenarios the technique can still perform well, however users of such a system should be encouraged to remain connected for longer to allow routing pathways to establish towards experts. Perhaps some form of incentive should be provided to encourage longer session durations (left for future work).

6.10 Attack Models

This section provides results for the more quantifiable attacks on the network and routing mechanisms. The random and stigmergic approaches are compared with the C1 (medium) churn scenario.

6.10.1 Eager Answer

With too many users in the network eagerly answering questions, without any expertise, problems appear within resulting answer qualities. Figure 6.58 provides the change in quality of the stigmergic approach in comparison to random when a proportion of the network is behaving in this manner. The particular form of attack makes it difficult to route questions as users are effectively prevent the routing from establishing. The stigmergic approach is still an improvement on the random approach. We also find that the V2 and V3 versions of the algorithm degrade more gracefully than the original stigmergic approach, strengthening the motivations for the alternative approaches.

6.10.2 False Feedback

The stigmergic routing relies on positive user feedback to learn about user expertise. When users in the network maliciously submit false positives there

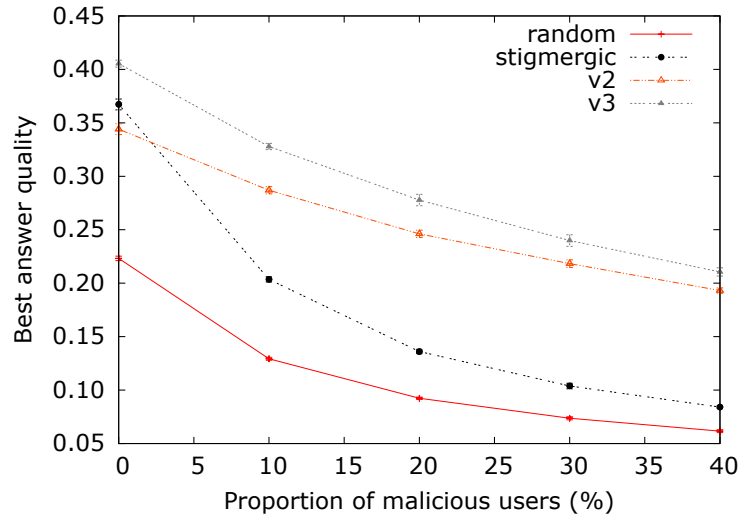


Figure 6.58: Eager answerer effect on quality.

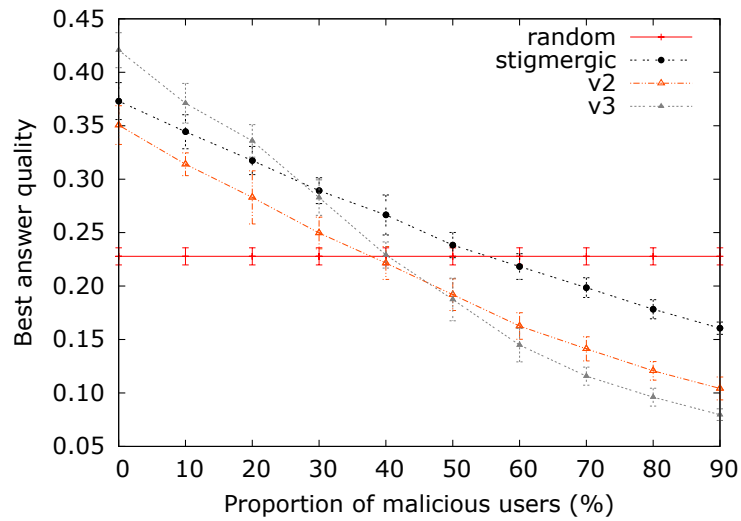


Figure 6.59: False feedback effect on quality.

is an obvious effect on answer quality. The routing will be tricked into routing question to non-expert users, Figure 6.59 provides results of various proportions of users maliciously submitting false positive feedback. As long as over fifty percent of the network are providing correct user feedback, the approach is still an improvement on the random approach.

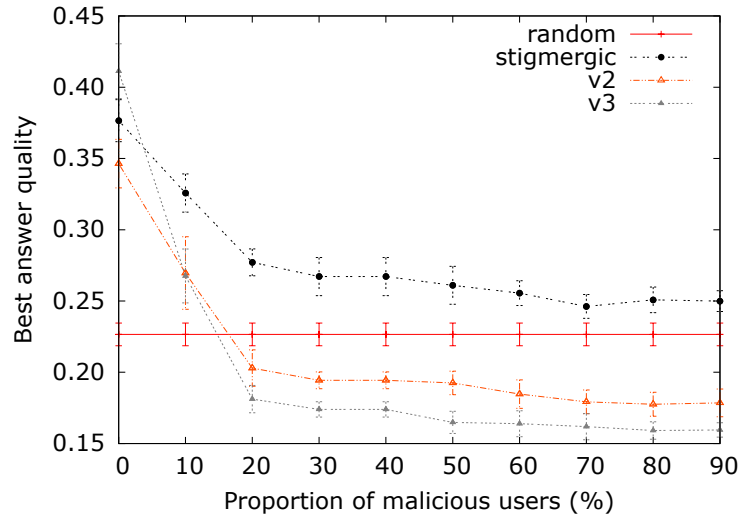


Figure 6.60: Question blocking effect on quality.

6.10.3 Question Blocker

Users may stop propagating questions when they are received as a form of attack against the routing protocol. Figure 6.60 shows the effect on answer quality as the proportion of malicious users using this attack rises.

6.10.4 Answer Blocker

Another form of attack would be blocking the propagation of answers back towards the original question askers. Figure 6.61 presents the results of various levels of malicious users using this form of attack.

6.10.5 Feedback Blocker

Finally, the feedback messages may be blocked by malicious users. The effect on quality can be seen in Figure 6.62 where we can see that the increase in malicious users reduces answer quality.

Overall the routing copes well with blocking and false feedback attacks, requiring a large proportion of the network to be misbehaving. However, if users prevent routing by always answering questions they have no expertise in – quality will quickly reduce for all non-flooding based approaches. If users are behaving in this manner, it may also block flooding for propagating questions

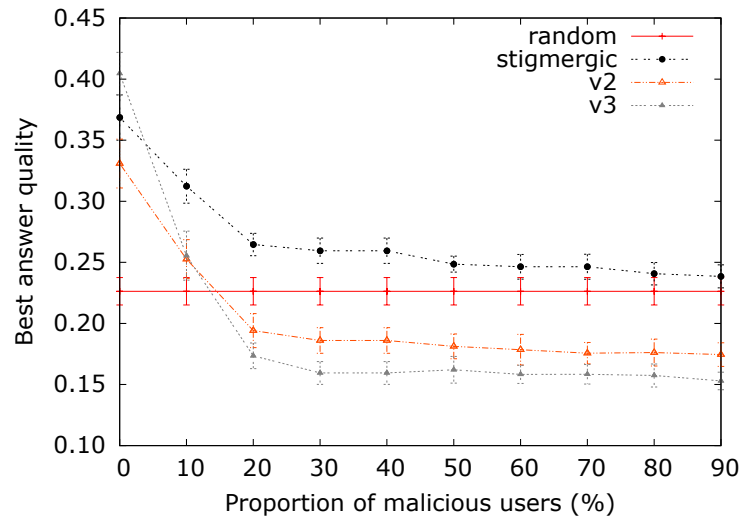


Figure 6.61: Answer blocking effect on quality.

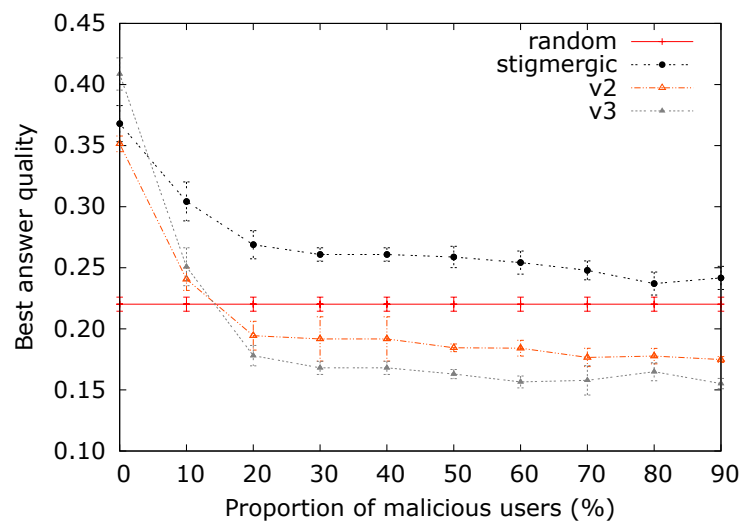


Figure 6.62: Feedback blocking effect on quality.

into the network. To route questions around the network in the decentralised setup, routing between nodes must not be prevented.

Conclusions & Future Work

The main aim of this thesis was to investigate, design and evaluate the concept of a distributed Q&A service which hides source identities within a crowd while striving to provide a fair service. The system developed is designed to be robust and adaptive which is an important element of the contribution.

With a clear and motivated research problem identified, a new approach towards question routing within ad hoc Q&A networks was presented. Experimental evaluation has found that it is not only possible to improve answer quality, but that the total amount of attention required to locate answers can be controlled and deniability for participants can be maintained. Although naïve approaches work well in small networks, careful consideration is needed when it comes to larger emergent networks in order to manage the workload of users and the quality of answers produced.

In order to balance answer quality and user attention we must ensure that we do not bombard experts in the network with questions. If we rely solely on the highest-ranking experts we will be disappointed with the level of unanswered questions and attention requirements on those few members of the community will be high. We cannot realistically expect to use a flooding approach, which consumes the maximum levels of attention, so we must opt for a more elegant autonomous option such as the stigmergic-inspired routing approach presented. We can achieve a good balance of attention workload with this technique, as it can be used to throttle requests, promote exploration and makes use of preferential expert selection.

7.1 General Observations & Lessons Learned

The evaluation of the routing approaches shows that a deniable and distributed Q&A network is possible, as long as we can encourage longer session durations and promote interactions within the network. As users become increasingly connected throughout their lives, control over authorship, logging and archiving are likely to become increasingly important issues, especially in the realm of human-orientated services such as Q&A.

The stigmergic approach towards deniable question routing is a valid and appropriate one. If users can be encouraged to participate and remain connected to the network for several hours or days then the routing will have time to establish and provide suitable levels of answer quality. The pheromone based probabilistic routing has the ability to learn and adapt as users come and go without the need for a central control or administration. This is particularly advantageous in distributed P2P environments which are notoriously difficult to moderate and control. It is the case that the more users participate with the service and the longer they remain connected the better the performance the networked crowd of individuals will receive.

Analysing a large collection of users as seen in this thesis is a long process which requires particular care and attention to detail. Designing and simulating scenarios involving a question and answer network enables various routing options to be considered and evaluated to verify the technique (such as the metrics and evaluation presented in this thesis).

In regards to the lessons learnt during this thesis, it would seem that nature has some very powerful and remarkably simple techniques which can be used efficiently for routing within communication networks. One should never underestimate the practicalities of designing and simulating a complex system. Expect to invest a large amount of time and effort to complete it. This work has created a large number of possible additional research questions and topics which are explored in the Future Work section below.

7.2 Future Work

The research area of decentralised Q&A systems presents many new and interesting challenges. Although outside the scope of this thesis, they are important and significant considerations to the work presented here.

7.2.1 Broken Routes

In networks exhibiting churn the route between question asker and answerers can be broken during the Q&A interaction. This is an issue as time and effort are consumed to generate any answer and if the chain is broken, this becomes futile. In this thesis, such occurrences are equivalent to an unanswered question, yet they still consume attention.

To solve broken routes, additional overlays could be created to allow answers to find alternative routes back to question askers. Emergent routes which aim to make connections towards nodes that have existed in the network for some time may be suitable here. The ability to route questions to a specific area of the network may allow answers to be forwarded or picked up later as an alternative answer delivery option.

7.2.2 Incentives

The longer users remain connected and actively participating in the Q&A network the better. As question asking and answering is deniable it becomes problematic to provide incentives for participation. Anonymous reputation-based systems could be used to allow some user controlled statistics, payments or verifications to be managed. Such systems have been investigated at the University of Sussex [96].

7.2.3 Real World Implementation

A real world implementation of the routing protocol would be an extremely interesting activity to investigate the way in which such a system is used, the behaviour of users and how the protocol performs in the real world. Such an implementation would be difficult to quantify however, it would allow for comparisons to be made between the simulation results presented in this thesis and the real software usage.

7.2.4 Routing Variations

There is the potential for a huge number of variations on the proposed stigmergic routing approach to question routing. It would be interesting for researchers to take and adapt the proposed routing to achieve additional goals, for example the duplication, replication and evolution of the various protocol messages


to achieve greater service robustness volatile networks or with greater model complexities and variations.

Appendix

Aardvark Anonymous

29/04/2010

<http://community.vark.com/forums/18354-general/suggestions/231239-allow-for-users-to-be-anonymous-at-times-i-do-no?ref=title>

 Aardvark

Join now!Aardvark on iPhoneSign In

Aardvark Community Forum

Everything we do at Aardvark is driven by the feedback we get from you, our users. So please add your ideas for new features, bugs to fix, or anything else into the Feedback Forum below -- or you can vote on ideas that other people have entered that you think are great. To keep up with the latest developments from the Aardvark team, check out [our blog](#), or become a fan of our [facebook page](#).

[« return to General Forum](#)

1,316
votes

vote


allow for users to be anonymous. at times I do not want to send specific response archived under me
don't want to have one username. would like an anonymous username i can use as well.

by anonymous | 139 comments

Status: under review

Aardvark is about real people interacting with real people, which is why we ask everyone to use their real names--just like on Facebook.

That said, we understand that there are personal topics that come up, in which case it might be preferred to ask or answer anonymously. I'd love to hear what other people on the forum think about this topic.

 **Alison**
admin

25 votes left!
[What happens if I run out?](#)

1st ranked

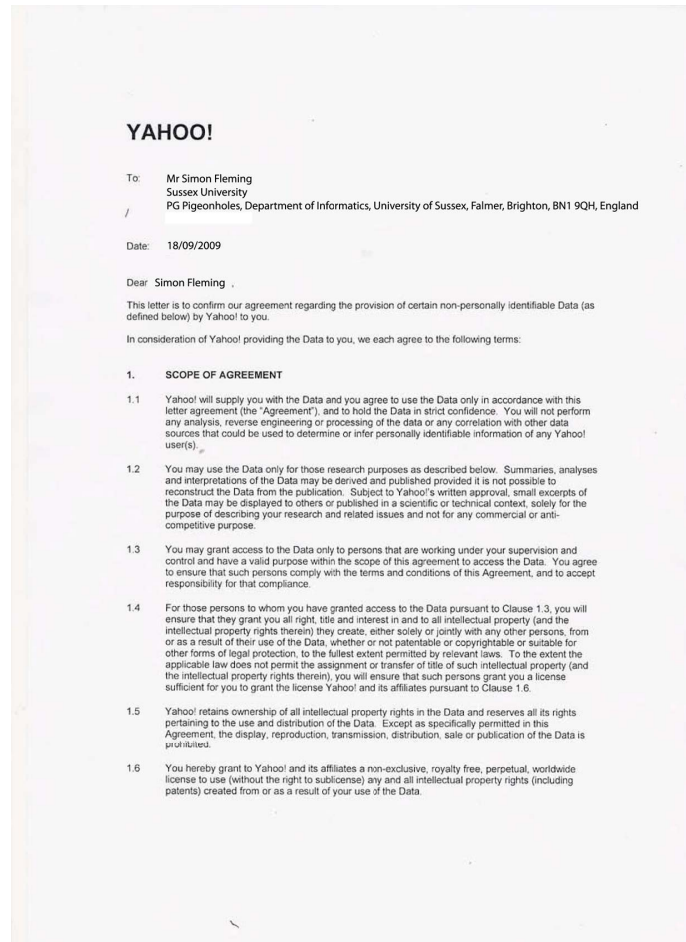
average votes
2.38

supporters
552

comments

[« Newer](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [Older](#) »

Yahoo Data Agreement



Corpus Structure

As with typical XML documents, the dataset starts with a root node, which in this case is called `<ytfed>` and acts as a wrapper for all questions contained in the corpus. This root node then contains each question within a `<vespaadd>` tag which then contains the following tags and associated data:

`<uri>`

The unique anonymised Universal Resource Identifier (URI).

<subject>

The actual question being asked.

<content>

Optional additional information about this question.

<bestanswer>

The answer selected as being the best for this question.

<nbestanswers>

All answers submitted in response to the question under concern are provided in unique **<answer_item>** elements.

<cat>

The actual category this question is assigned to.

<maincat>

The main category assigned to this question.

<subcat>

The sub-category assigned to this question.

<id>

The anonymized ID of the user who asked the question.

<best_id>

The anonymized ID of the user who provided the best answer.

<qlang>

Indicates the language the Q&A were posted in.

<qint1>

The location from which the question was posted.

<date>

The unix time stamp of when the question was first created.

<lastanswers>

The unix time stamp of the last answer given for the question.

<res_date>

The unix time stamp of when the question was resolved (best answer confirmed).

<vot_date>

The unix time stamp of the best answer vote.

Example Simulation Configuration File

Listing 8.1: simulation configuration file.

```
POP_SIZE=100000
SIM_LEN=50000
INIT_SIZE=1000
QRATE=0.000175

WEI_K=3.07
WEI_LAMBDA=269.79
MIN_STAY_DURATION=1

TRANS_PROB_ATTENTION=0.9996
TRANS_PROB_IDLE=0.9996
REQ_ANS=5
QUEUE_SIZE=15
ITERATIONS=5

PHEROMONE_RATE=0.05
PHEROMONE_FEEDBACK=0.8
PHEROMONE DISSOLVE_RATE=0.001
DISSOLVE_MOD_STEPS=250
PHEROMONE_CAP=5
PHEROMONE_Q_RATE_V3=0.05
WARMUP_PERIOD=15000
STARTUP_PHEROMONE=1.0
DEFAULT_PHEROMONE_VALUE=0.06

SEED=-8433177374722890506
MOD_STEPS_GIVE_NETSIZE=180

FUNCTION_F=FALSE
FUNCTION_TO_RUN=EAGER
EXP_EXTRA=CHURN
```

Example Bash Script

Listing 8.2: bash script example.

```
# record some extra results
echo "..._right_..."
cat raw_stats/question_cats.dat | sort -n | uniq -c > raw_stats/cats.dat
cat raw_stats/expertise_values.dat | sort -n | uniq -c > raw_stats/expertise_values_pro.dat
# plot first level graphs
cd graphs
gnuplot *
rm *.gnu
mv ../convert.sh .
./convert.sh
```

```
rm *.eps

# plot second level ewma graphs
cd EWMA
ls *.gnu |xargs -t -I{} gnuplot {}
mv ../convert.sh .
./convert.sh
```

Example Gnuplot Script

Listing 8.3: gnuplot script example.

```
reset
set grid
set boxwidth 0.1 absolute
set terminal postscript eps enhanced color
set title 'Best_Answer_quality_percentiles:_5th,_25th,_Mean,_75th_and_95th'
set ylabel 'Answer_Quality'
set xlabel 'Question_Routing_Approach'
set output 'answer_quality_best.eps'
set xrange [ 0.00000 : 6.0000 ] noreverse nowriteback
set yrange [ 0.0 : 1.0 ]
set xtics (" 0, "Flooding" 1, "Random" 2, "Stigmergic" 3, "V2" 4, "V3" 5)
plot '../raw_stats/answer_quality_percentiles_best.dat' using 1:3:2:6:5
    with candlesticks lt 1 lw 2 title 'Quartiles' whiskerbars, \
'' using 1:4:4:4:4 with candlesticks lt -1 lw 2 notitle
```

Expertise per top level Yahoo! Answers category.

Pheromone Investigations

Name	Qrate	Answers	Attention Model	Churn
AA1	Low (0.0001)	1	Low (0.25 remain, 0.75 transition)	–
AA2	High (0.0003)	1	Low	–
BB1	Low	1	High (0.75 remain, 0.25 transition)	–
BB2	High	1	High	–
CC1	Low	1	Low	YES
CC2	High	1	Low	YES
DD1	Low	1	High	YES
DD2	High	1	High	YES

Table 8.1: Investigation details.

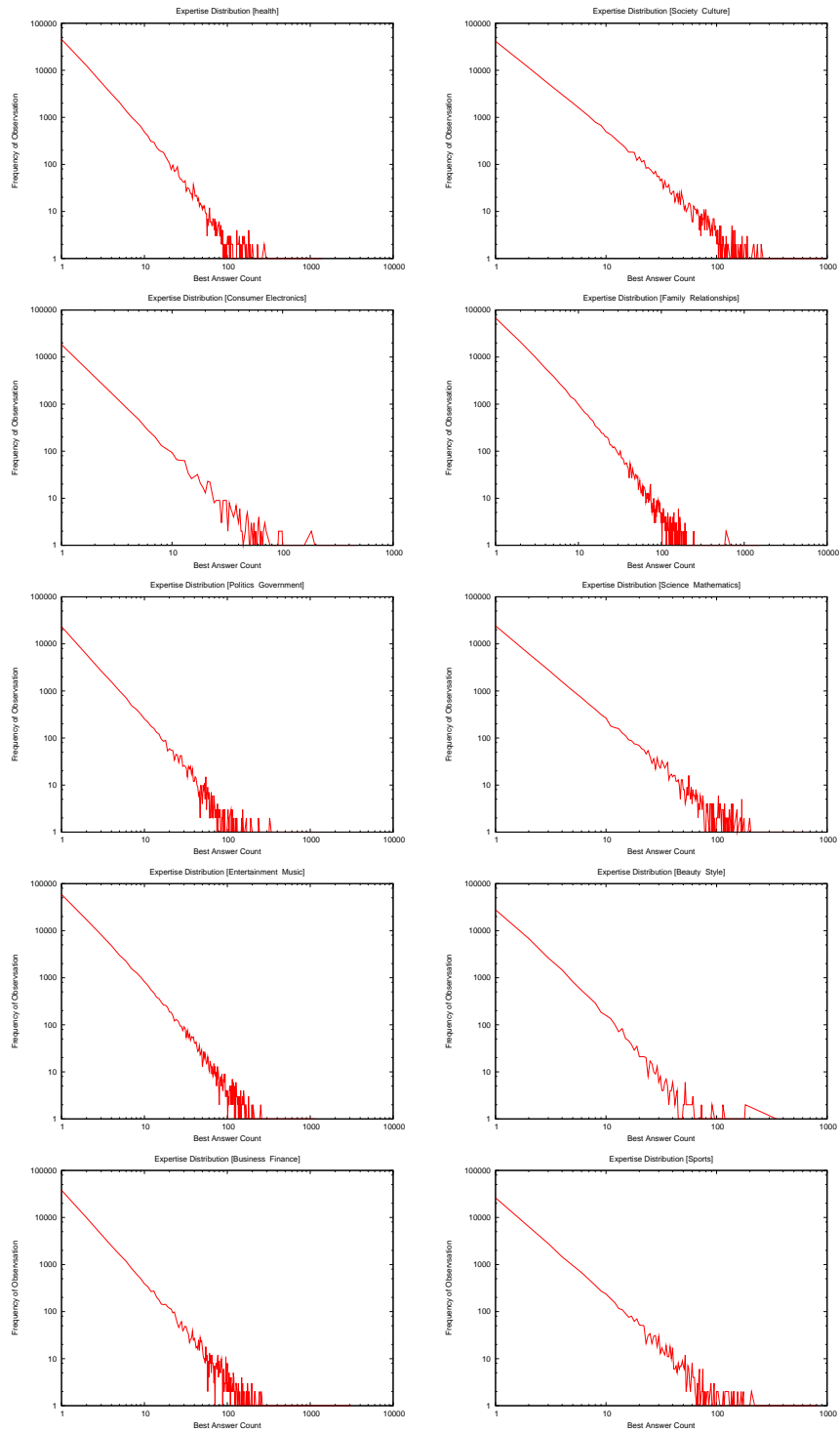


Figure 8.1: Yahoo! Answers Categories 1 to 10.

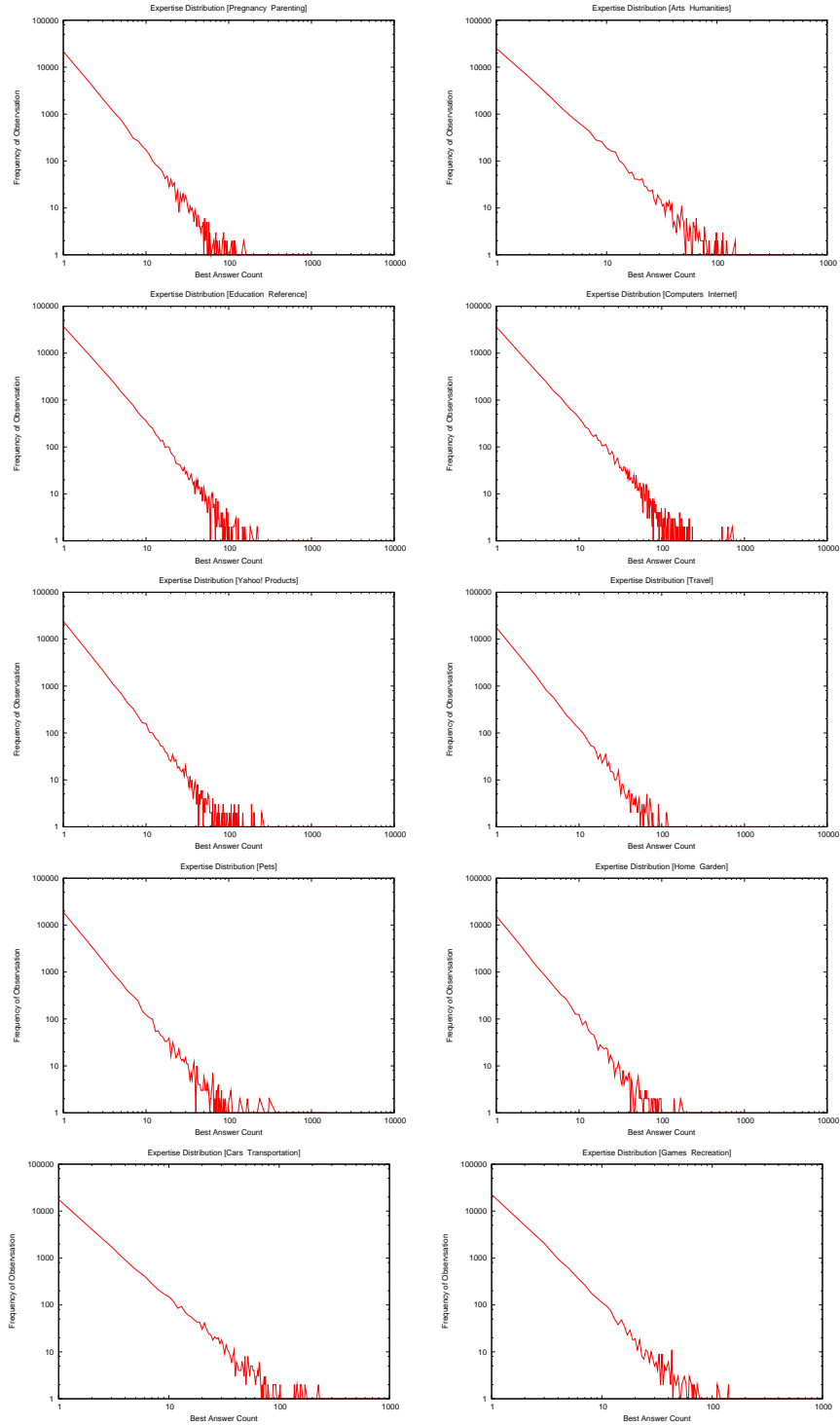


Figure 8.2: Yahoo! Answers Categories 11 to 20.

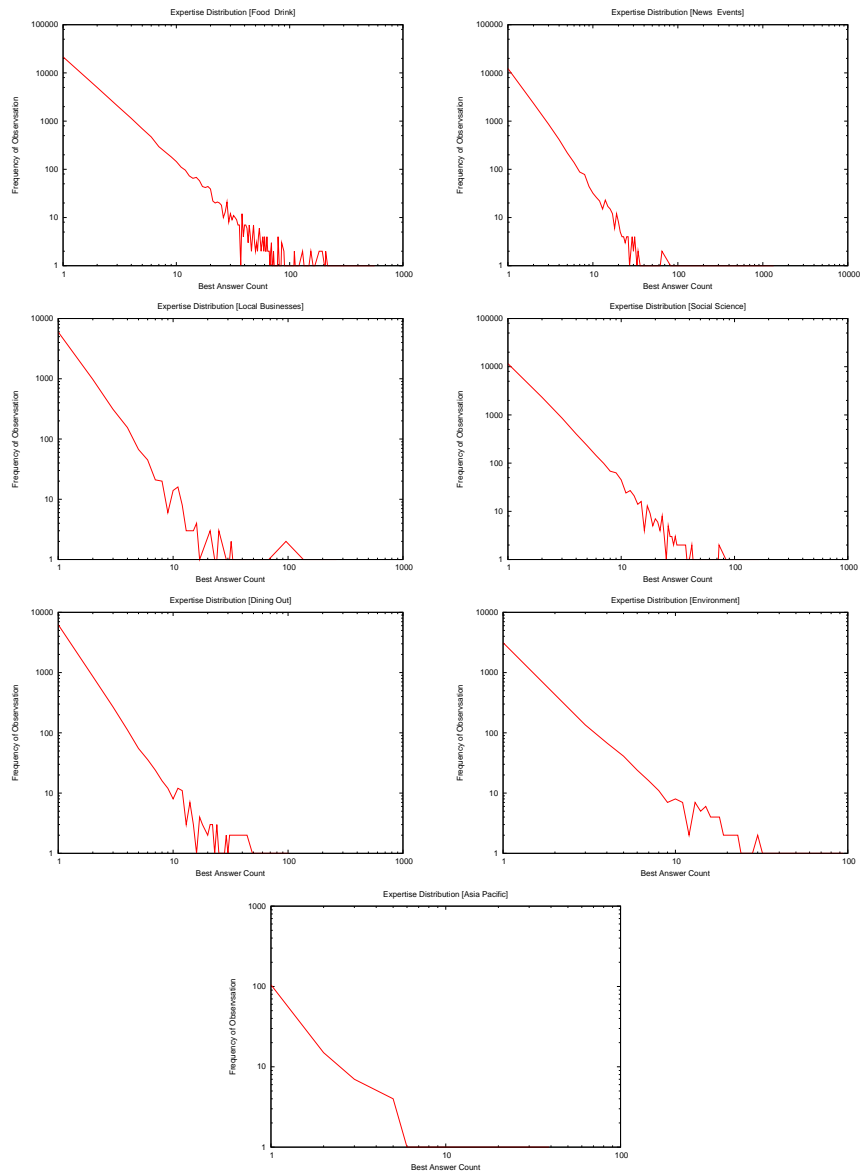


Figure 8.3: Yahoo! Answers Categories 21 to 27.

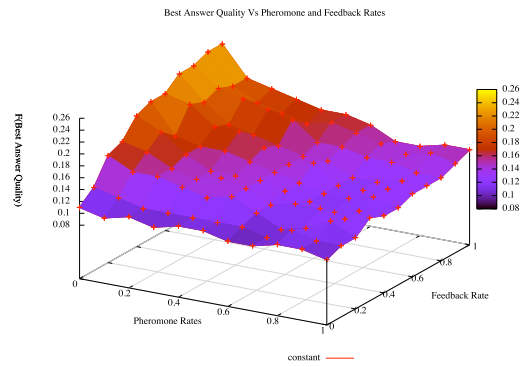


Figure 8.4: AA1 quality.

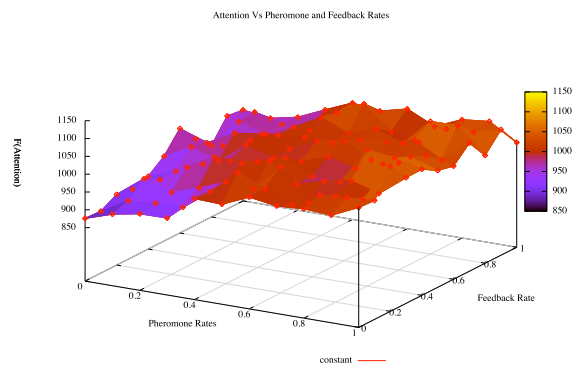


Figure 8.5: AA1 attention.

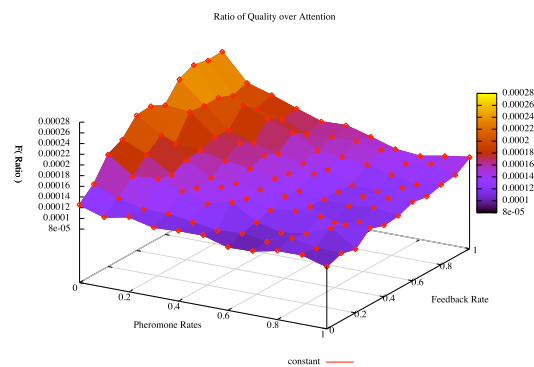


Figure 8.6: AA1 ratio.

Figure 8.7: Low Question Rate and Attention (AA1).

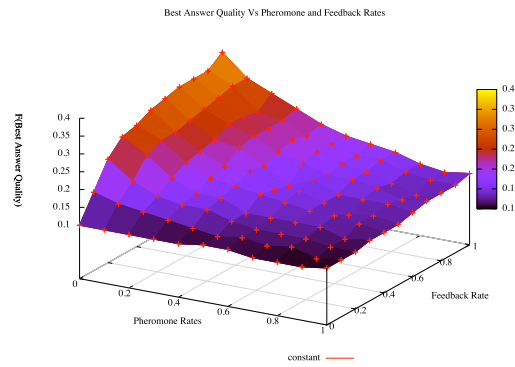


Figure 8.8: AA2 quality.

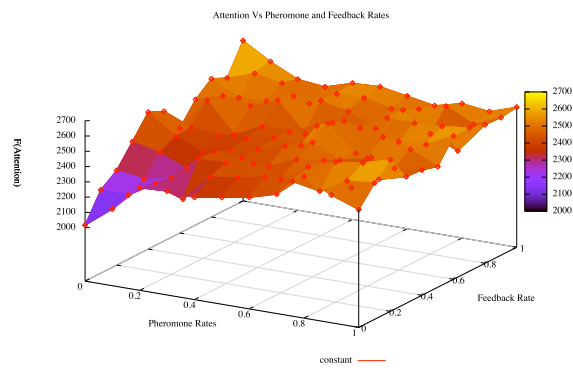


Figure 8.9: AA2 attention.

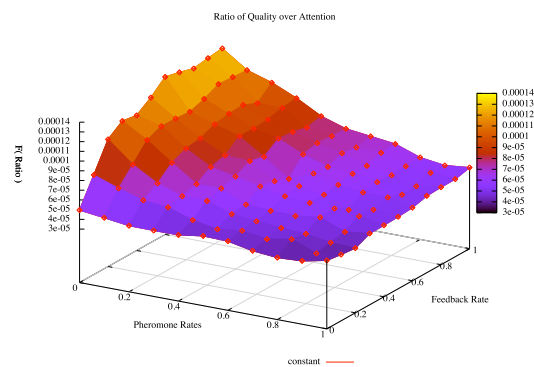


Figure 8.10: AA2 ratio.

Figure 8.11: High Question Rate with Low Attention (AA2).

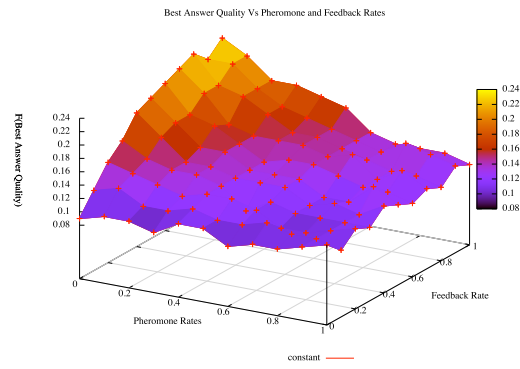


Figure 8.12: BB1 quality.

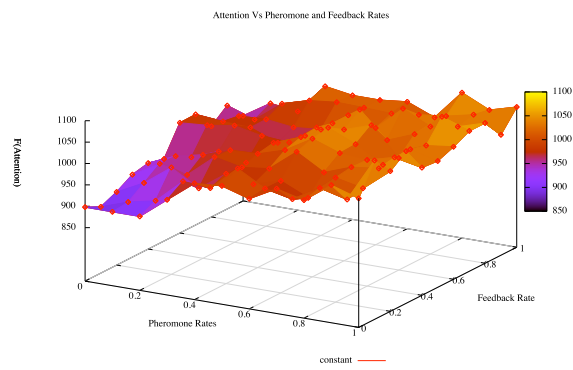


Figure 8.13: BB1 attention.

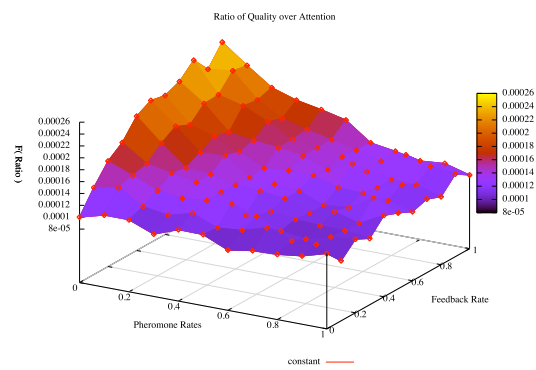


Figure 8.14: BB1 ratio.

Figure 8.15: Low Question Rate with High Attention (BB1).

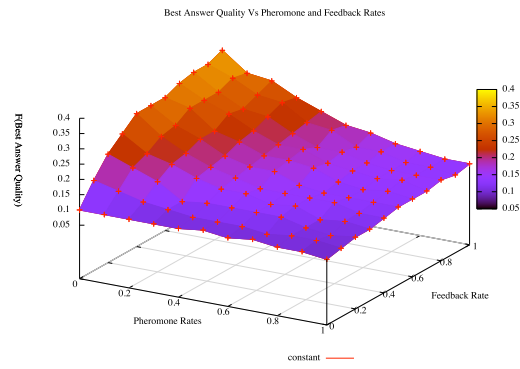


Figure 8.16: BB2 quality.

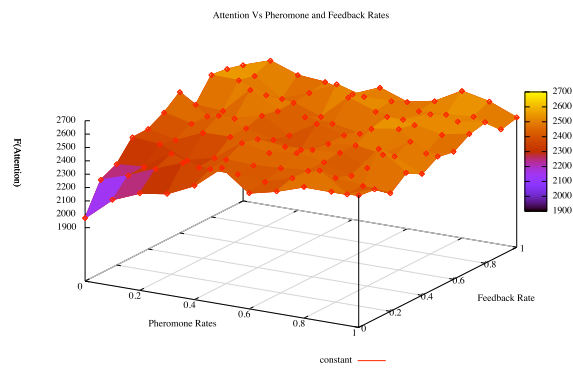


Figure 8.17: BB2 attention.

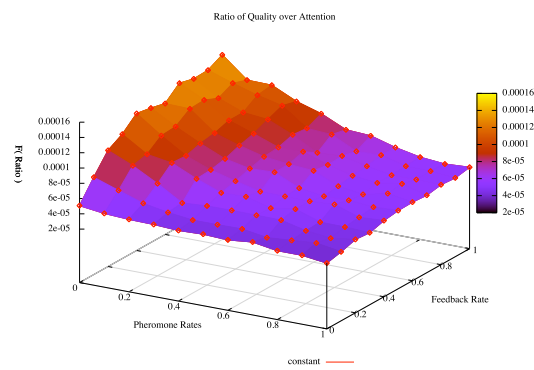


Figure 8.18: BB2 ratio.

Figure 8.19: High Question Rate with High Attention (BB2).

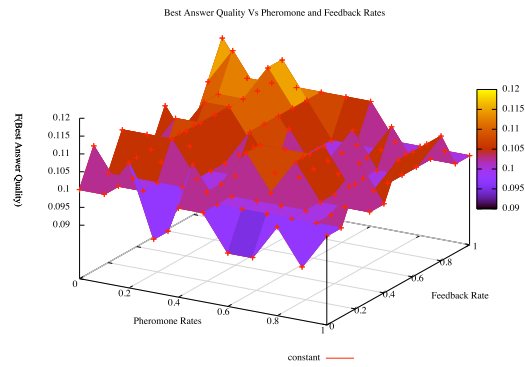


Figure 8.20: CC1 quality.

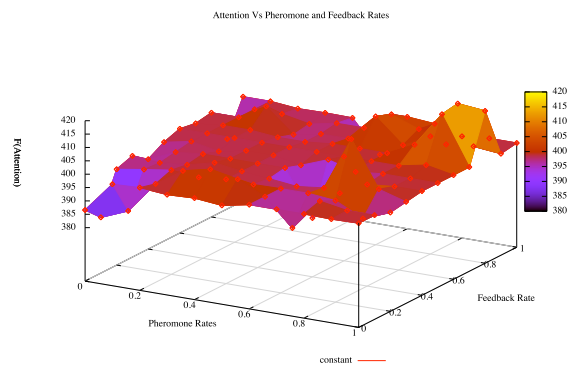


Figure 8.21: CC1 attention.

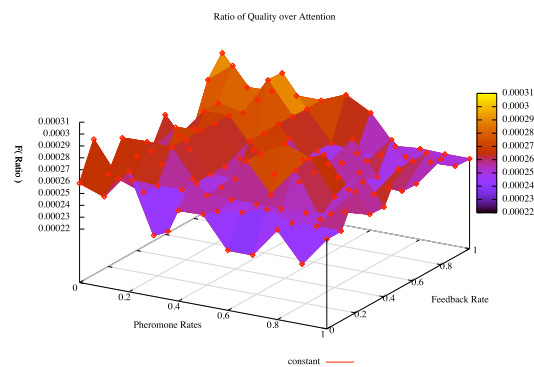


Figure 8.22: CC1 ratio.

Figure 8.23: Low Question Rate and Attention with Churn (CC1).

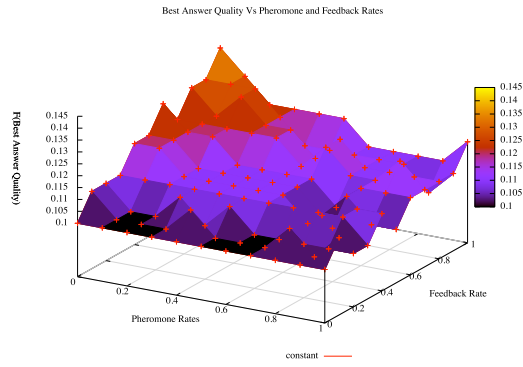


Figure 8.24: CC2 quality.

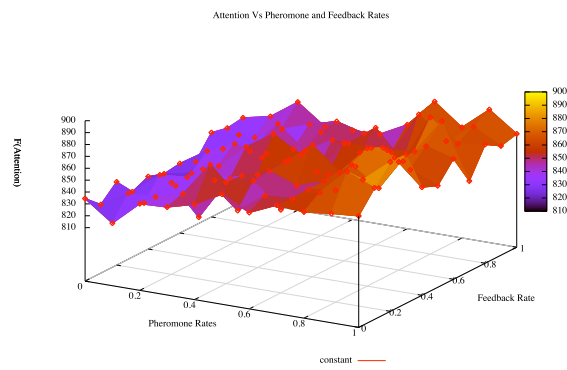


Figure 8.25: CC2 attention.

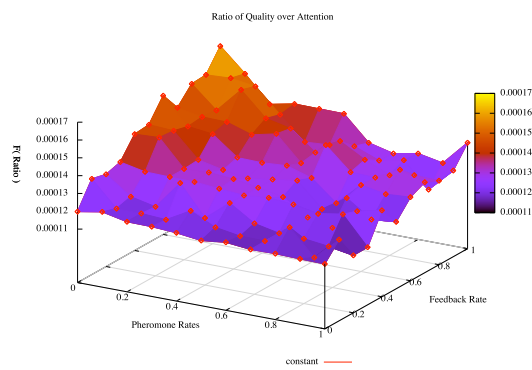


Figure 8.26: CC2 ratio.

Figure 8.27: Low Question Rate and Attention with Churn (CC2).

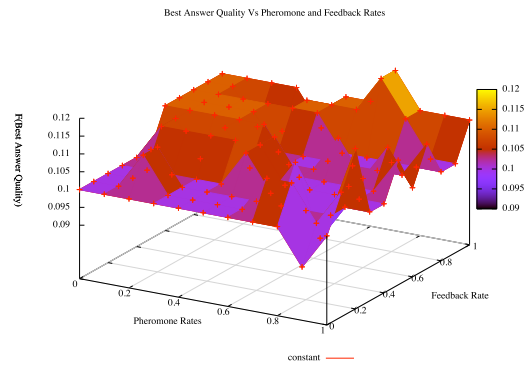


Figure 8.28: DD1 quality.

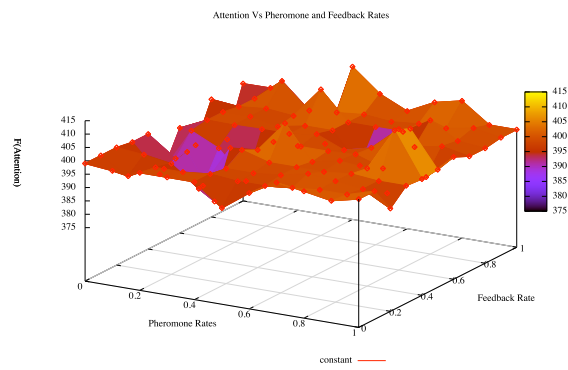


Figure 8.29: DD1 attention.

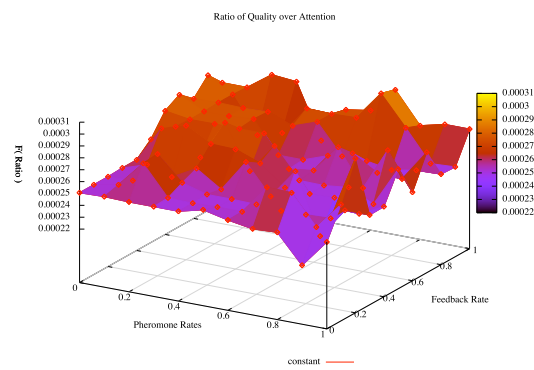


Figure 8.30: DD1 ratio.

Figure 8.31: Low Question Rate with High Attention and Churn (DD1).

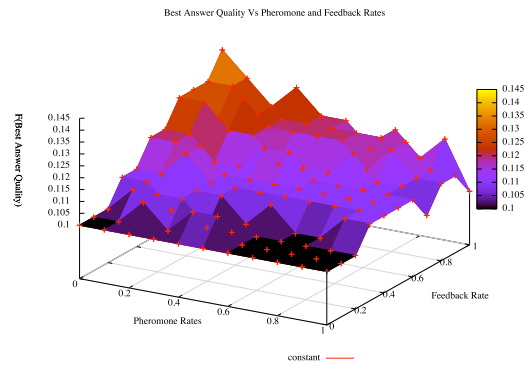


Figure 8.32: DD2 quality.

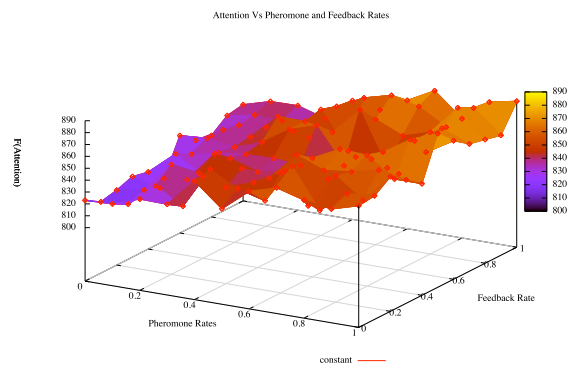


Figure 8.33: DD2 attention.

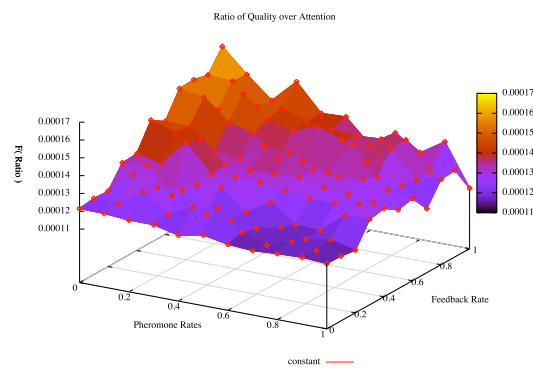


Figure 8.34: DD2 ratio.

Figure 8.35: High Question Rate with High Attention and Churn (DD2).

Bibliography

- [1] Damon Horowitz and Sepandar D. Kamvar. The anatomy of a large-scale social search engine. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 431–440, New York, NY, USA, 2010. ACM.
- [2] Fluther. URL <http://www.fluther.com/>. last accessed 19/09/2011.
- [3] David Dearman and Khai N. Truong. Why users of Yahoo! Answers do not answer questions. In *CHI '10: Proceedings of the 28th International Conference on Human Factors in Computing Systems*, pages 329–332, New York, NY, USA, 2010. ACM.
- [4] Michael K Reiter and Aviel D Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security*, pages 66–92, 1998.
- [5] Mouna Kacimi, Stefano Ortolani, and Bruno Crispo. Anonymous opinion exchange over untrusted social networks. In *SNS '09: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pages 26–32, New York, NY, USA, 2009. ACM.
- [6] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, pages 46–66, 2001.
- [7] Ruud Schoonderwoerd, Janet L. Bruten, Owen E. Holland, and Leon J. M. Rothkrantz. Ant-based load balancing in telecommunications networks. *Adaptive Behaviour*.
- [8] Devika Subramanian, Peter Druschel, and Johnny Chen. Ants and reinforcement learning: a case study in routing in dynamic networks.

- In *Proceedings of the Fifteenth international joint conference on Artificial intelligence - Volume 2*, pages 832–838, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-555860-480-4. URL <http://dl.acm.org/citation.cfm?id=1622270.1622276>.
- [9] Benjamín Barán and Rubén Sosa. Antnet: Routing algorithm for data networks based on mobile agents. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 5(12):75–84, 2001.
 - [10] M. Roth and S. Wicker. Termite: ad-hoc networking with stigmergy. In *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, volume 5, pages 2937–2941, 2003.
 - [11] M. Heissenbüttel, M. Heissenbüttel, T. Braun, and T. Braun. Ants-based routing in large scale mobile ad-hoc networks. In *Kommunikation in verteilten Systemen (KiVS03)*, pages 91–99, 2003.
 - [12] John S. Baras and Harsh Mehta. A probabilistic emergent routing algorithm for mobile ad hoc networks. In *WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, page 10, 2003.
 - [13] Haihua Yun and A. Nur Zincir-Heywood. Intelligent ants for adaptive network routing. In *Conference on Communication Networks and Services Research*, pages 255–261, 2004.
 - [14] Gianni Di Caro, Frederick Ducatelle, and Luca Maria Gambardella. Anthocnet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, 16:443–455, 2005.
 - [15] Dan Cosley, Dan Frankowski, Loren G. Terveen, and John Riedl. Suggestbot: using intelligent task routing to help people find work in Wikipedia. In *Intelligent User Interfaces*, pages 32–41, 2007.
 - [16] Gianluca Demartini. Finding experts using Wikipedia. In *Finding Experts on the Web with Semantics*, pages 33–41, 2007.
 - [17] Yanhong Zhou, Gao Cong, Bin Cui, Christian S. Jensen, and Junjie Yao. Routing questions to the right users in online communities. In *International Conference on Data Engineering*, pages 700–711, 2009.

- [18] Craig Macdonald, David Hannah, and Iadh Ounis. High quality expertise evidence for expert search. In *European Colloquium on IR Research*, pages 283–295, 2008.
- [19] Nick Craswell, David Hawking, Anne-Marie Vercoustre, and Peter Wilkins. P@noptic expert: Searching for experts not just for documents. In *Ausweb*, pages 21–25, 2001.
- [20] Xiaoyong Liu, W. Bruce Croft, and Matthew Koll. Finding experts in community-based question-answering services. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 315–316, New York, NY, USA, 2005. ACM.
- [21] Craig Macdonald and Iadh Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 387–396, New York, NY, USA, 2006. ACM.
- [22] Apple Inc. Apple (United Kingdom) - iPad - the best way to experience the web, email & photos, 2011.
- [23] M Weiser. The computer for the 21st century. *Scientific American*, 3: 94–104, 1991.
- [24] Brian H. Murray and Alvin Moore. Sizing the internet. White paper, Cyveillance, July 2000. URL http://www.cyveillance.com/web/us/downloads/Sizing_the_Internet.pdf.
- [25] T. Berners-Lee and D. Connolly. *Hypertext Markup Language - 2.0*. United States, 1995.
- [26] M. Murata, S. St. Laurent, and D. Kohn. *XML Media Types*. United States, 2001.
- [27] Jędrzej Rybicki, Björn Scheuermann, Wolfgang Kiess, Christian Lochert, Pezhman Fallahi, and Martin Mauve. Challenge: peers on wheels - a road to new traffic information systems. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 215–221, New York, NY, USA, 2007. ACM.

- [28] Courtenay Honeycutt and Susan C. Herring. Beyond microblogging: Conversation and collaboration via twitter. *Hawaii International Conference on System Sciences*, pages 1–10, 2009.
- [29] Nick Saint. Twitter visualization shows when we are happiest, 2010. URL <http://www.sfgate.com/cgi-bin/article.cgi?f=/g/a/2010/07/21/businessinsider-what-twitter-says-about-how-happy-we-are-2010-7.DTL>.
- [30] R. Feizy, I. Wakeman, and D. Chalmers. Distinguishing fact and fiction: Data mining online identities. In *STM 2009: Proceedings of 5th International Workshop on Security and Trust Management*, 2009.
- [31] R. Feizy, I. Wakeman, and D. Chalmers. The transformation of online representation through time in relations to honesty and accountability characteristics. In *Proceedings of Advances in Social Network Analysis and Mining (ASONAM 2009)*, 2009.
- [32] Tim O'Reilly and John Battelle. Web squared: Web 2.0 five years on, 2009. URL <http://www.web2summit.com/web2009/public/schedule/detail/10194>.
- [33] Jacob Palme. You have 134 unread mail! do you want to read them now? In *Proceedings of the IFIP WG 6.5 working conference on Computer-based message services*, pages 175–184, New York, NY, USA, 1984. Elsevier North-Holland, Inc.
- [34] Eric Horvitz and Johnson Apacible. Learning and reasoning about interruption. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 20–27, New York, NY, USA, 2003. ACM.
- [35] Grace YoungJoo Jeon, Yong-Mi Kim, and Yan Chen. Re-examining price as a predictor of answer quality in an online q&a site. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 325–328, New York, NY, USA, 2010. ACM.
- [36] Gary Hsieh, Robert E. Kraut, and Scott E. Hudson. Why pay? exploring how financial incentives are used for question & answer. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 305–314, New York, NY, USA, 2010. ACM.

- [37] F. Maxwell Harper, Daniel Moy, and Joseph A. Konstan. Facts or friends? distinguishing informational and conversational questions in social q&a sites. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 759–768, New York, NY, USA, 2009. ACM.
- [38] Zoltan Gyongyi, Georgia Koutrika, Jan Pedersen, and Hector Garcia-Molina. Questioning Yahoo! answers. Technical Report 2007-35, Stanford InfoLab, 2007.
- [39] Gary Hsieh and Scott Counts. mimir: a market-based real-time question and answer service. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 769–778, New York, NY, USA, 2009. ACM.
- [40] Ian Wakeman, Dan Chalmers, and Michael Fry. Reconciling privacy and security in pervasive computing: the case for pseudonymous group membership. In *MPAC '07: Proceedings of the 5th international workshop on Middleware for pervasive and ad-hoc computing*, pages 7–12, New York, NY, USA, 2007. ACM.
- [41] Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani, Sabrina De Capitani di Vimercati, and Pierangela Samarati. Location privacy protection through obfuscation-based techniques. In *DBSec*, volume 4602 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2007.
- [42] David Evans, Alastair R. Beresford, Trevor Burbridge, and Andrea Soper. Context-derived pseudonyms for protection of privacy in transport middleware and applications. In *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, Washington, DC, USA, 2007. IEEE Computer Society.
- [43] Oliver Jorns, Gerald Quirchmayr, and Oliver Jung. A privacy enhancing mechanism based on pseudonyms for identity protection in location-based services. In *ACSW '07: Proceedings of the fifth Australasian symposium on ACSW frontiers*, pages 133–142, Darlinghurst, Australia, 2007. Australian Computer Society, Inc.

- [44] Ralph C. Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21(4):294–299, 1978.
- [45] *RFC 791 Internet Protocol - DARPA Internet Programm, Protocol Specification*. Internet Engineering Task Force, September 1981.
- [46] S. Deering and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*. United States, 1998.
- [47] Stuart Staniford, Vern Paxson, and Nicholas Weaver. How to own the Internet in your spare time. In *Proc. 11th USENIX Security Symposium*, San Francisco, CA, August 2002.
- [48] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.
- [49] Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. Anonymous connections and onion routing. In *SP '97: Proceedings of the 1997 IEEE Symposium on Security and Privacy*, Washington, DC, USA, 1997. IEEE Computer Society.
- [50] A Pfitzmann and M Waidner. Networks without user observability. *Computer Security*, 6(2):158–166, 1987.
- [51] Jochen H. Schiller and Agnès Voisard. *Location-Based Services*. Morgan Kaufmann, 2004.
- [52] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42, New York, NY, USA, 2003. ACM.
- [53] Chi-Yin Chow, Mohamed F. Mokbel, and Xuan Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *GIS '06: Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 171–178, New York, NY, USA, 2006. ACM.
- [54] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*. Oxford University Press, Inc., New York, NY, USA, 1999.

- [55] J. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels. The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, 3: 159–168, March 1990.
- [56] S. Goss, S. Aron, J. Deneubourg, and J. Pasteels. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76(12):579–581, December 1989.
- [57] Marco Dorigo, Vittorio Maniezzo, Alberto Coloni, and Piazza Leonardo da Vinci. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26:29–41, 1996.
- [58] Marco Dorigo and Luca Maria Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1:53–66, 1997.
- [59] Van Dyke. Expert Assessment of Human-Human Stigmergy. Technical report, Altarum Institute, 3520 Green Court, Suite 300. Ann Arbor, Michigan 48105, May 2005.
- [60] Laura Rosati, Matteo Berlioli, and Gianluca Reali. On ant routing algorithms in ad hoc networks with critical connectivity. *Ad Hoc Networks*, 6:827–859, August 2008.
- [61] Elke Michlmayr, Sabine Graf, Wolf Siberski, and Wolfgang Nejdl. Query routing with ants. In *ESWC2005: Proceedings of the 1st Workshop on Ontologies in P2P Communities*, 2005.
- [62] Elke Michlmayr. Self-organization for search in peer-to-peer networks: the exploitation-exploration dilemma. In *BIONETICS '06: Proceedings of the 1st international conference on Bio inspired models of network, information and computing systems*, New York, NY, USA, 2006. ACM.
- [63] Matteo Berlioli, Laura Rosati, Markus Werner, Gianluca Reali, Matteo Berlioli, Laura Rosati, Markus Werner, and Gianluca Reali. Ant routing concepts for dynamic meshed satellite constellations, 2004.
- [64] Cheng-Chang Hoh, Chiung-Ying Wang, and Ren-Hung Hwang. Anycast routing protocol using swarm intelligence for ad hoc pervasive network.

- In *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, IWCMC '06, pages 815–820, New York, NY, USA, 2006. ACM.
- [65] Alfredo Garcia and Fernan A. Pedraza. Rational swarm routing protocol for mobile ad-hoc wireless networks. In *Proceedings of the 5th international conference on Pervasive services*, ICPS '08, pages 21–26, New York, NY, USA, 2008. ACM.
- [66] Floriano De Rango and Mauro Tropea. Swarm intelligence based energy saving and load balancing in wireless ad hoc networks. In *BADS '09: Proceedings of the 2009 workshop on Bio-inspired algorithms for distributed systems*, pages 77–84, New York, NY, USA, 2009. ACM.
- [67] John Risson and Tim Moors. Survey of research towards robust peer-to-peer networks: search methods. *Computer Networks*, 50:3485–3521, December 2006. ISSN 1389-1286.
- [68] Daniel Stutzbach and Reza Rejaie. Understanding churn in peer-to-peer networks. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 189–202, New York, NY, USA, 2006. ACM.
- [69] W. J. Croft and J. Gilmore. RFC 951: Bootstrap protocol, September 1985.
- [70] Christoph Tempich, Steffen Staab, and Adrian Wranik. Remindin': semantic query routing in peer-to-peer networks based on social metaphors. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 640–649, New York, NY, USA, 2004. ACM.
- [71] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, New York, NY, USA, 2001. ACM.
- [72] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In

- SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 267–278, New York, NY, USA, 2006. ACM.
- [73] S. Naicken, B. Livingston, A. Basu, S. Rodhetbhai, I. Wakeman, and D. Chalmers. The State of Peer-to-Peer Simulators and Simulations. *ACM Computer Communications Review*, 37(2):95–98, 2007.
- [74] Yahoo! Webscope Datasets Catalog (L6). Yahoo! answers comprehensive questions and answers version 1.0, 2009. URL <http://www.stanford.edu/class/cs345a/YahooData.pdf>.
- [75] Albert-Laszlo Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207–211, May 2005.
- [76] Jon Kleinberg. Bursty and hierarchical structure in streams. pages 91–101, 2002.
- [77] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. Analysis of text entry performance metrics. In *Proceedings. IEEE TIC-STH 2009. IEEE*, pages 100–105, 2009.
- [78] Clare-Marie Karat, Christine Halverson, Daniel Horn, and John Karat. Patterns of entry and correction in large vocabulary continuous speech recognition systems. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 568–575, New York, NY, USA, 1999. ACM.
- [79] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E. W. Looney. Twiddler typing: one-handed chording text entry for mobile phones. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 671–678, New York, NY, USA, 2004. ACM.
- [80] Mackenzie and Soukoreff. Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*, 17:147–198, 2002.
- [81] Martina Ziefle. Effects of display resolution on visual performance. *Human Factors*, 40(4):554–568, 1998.

- [82] Márk Jelasity and Alberto Montresor. Epidemic-style proactive aggregation in large overlay networks. In *ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems*, pages 102–109, Washington, DC, USA, 2004. IEEE Computer Society.
- [83] Daniel Stutzbach and Reza Rejaie. Understanding churn in peer-to-peer networks. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 189–202, New York, NY, USA, 2006. ACM.
- [84] Zhen Xiao, Lei Guo, and John Tracey. Understanding instant messaging traffic characteristics. In *ICDCS '07: Proceedings of the 27th International Conference on Distributed Computing Systems*, Washington, DC, USA, 2007. IEEE Computer Society.
- [85] Raj Jain. *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley, 1991.
- [86] P. Erdős and A. Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.
- [87] Benjamín Barán. Improved antnet routing. *SIGCOMM Comput. Commun. Rev.*, 31:42–48, April 2001.
- [88] Shlomo Argamon, Marin Šarić, and Sterling S. Stein. Style mining of electronic messages for multiple authorship discrimination: first results. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 475–480, New York, NY, USA, 2003. ACM.
- [89] Hugo Jair Escalante, Thamar Solorio, and Manuel Montes-y Gómez. Local histograms of character n-grams for authorship attribution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 288–298, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
- [90] PlanetSim Core Team. PlanetSim: Object oriented simulation framework for overlay networks, 2009. URL <http://projects-deim.urv.cat/trac/planetsim/>.

- [91] Alberto Montresor and Mark Jelasity. Peersim: A scalable p2p simulator. In *Peer-to-Peer Computing, 2009. P2P'09. IEEE Ninth International Conference on*, pages 99–100. IEEE, September 2009.
- [92] T.M. Gil, F. Kaashoek, J. Li, R. Morris, and J. Stribling. p2psim: a simulator for peer-to-peer (p2p) protocols, 2005. URL <http://pdos.csail.mit.edu/p2psim/>.
- [93] Kazayuki Shudo, Y. Tanaka, and S. Sekiguchi. Overlay weaver: An overlay construction toolkit, 2008. ISSN 0140-3664.
- [94] Ingmar Baumgart, Bernhard Heep, and Stephan Krause. OverSim: A Flexible Overlay Network Simulation Framework. *2007 IEEE Global Internet Symposium*, pages 79–84, may 2007.
- [95] Mary Natrella. *NIST/SEMATECH e-Handbook of Statistical Methods*. NIST/SEMATECH, 2010. URL <http://www.itl.nist.gov/div898/handbook/>.
- [96] A. Basu, I. Wakeman, and D. Chalmers. A Framework for Developing and Sharing Client Reputations. In *Proceedings of the Fourth IFIP WG 11.11 International Conference on Trust Management (IFIPTM), Morioka, Japan*, 2010.